CDRL: B001
25 March 1994

UNISYS

Market Study

Central Archive for Reusable Defense Software (CARDS)

Informal Technical Data

CARDS

Central Archive for Reusable Defense Software

STARS-VC-B001/004/00
25 March 1994

CDRL: B001
25 March 1994

INFORMAL TECHNICAL REPORT
For The
SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS
(STARS)

Market Study
Central Archive for Reusable Defense Software
(CARDS)

STARS-VC-B001/004/00
25 March 1994

Data Type: Informal Technical Data

CONTRACT NO. F19628-93-C-0130
Line Item 0002AB

Prepared for:

Electronic Systems Center
Air Force Material Command, USAF
Hanscom AFB, MA 01731-2816

Prepared By:

DSD Laboratories, Inc.
under contract to
Unisys Corporation
12010 Sunrise Valley Drive
Reston VA 22091

Distribution Statement "A"
per DoD Directive 5230.24
Approved for public release; Distribution is unlimited

CDRL: B001
25 March 1994

Data Reference: STARS-VC-B001/004/00
INFORMAL TECHNICAL REPORT
Market Study
Central Archive for Reusable Defense Software (CARDS)

Distribution Statement "A"
per DoD Directive 5230.24
Approved for public release; Distribution is unlimited.

This document, developed under the Software Technology for Adaptable, Reliable Systems (STARS) program, is approved for release under Distribution "A" of the Scientific and Technical Information Program Classification Scheme (DoD Directive 5230.24) unless otherwise indicated. Sponsored by the U. S. Advanced Research Projects Agency (ARPA) under contract F19628-93-C-0130 the STARS program is supported by the military services, SEI, and MITRE, with the U. S. Air Force as the executive contracting agent. The information identified herein is subject to change. For further information, contact the authors at the following mailer address: delivery@stars.reston.paramax.com.

CDRL: B001
25 March 1994

Data Reference: STARS-VC-B001/004/00
INFORMAL TECHNICAL DOCUMENT
Market Study
Central Archive for Reusable Defense Software (CARDS)

Principal author of update:

Lynn G. Meheran

Approvals:

System Architect ***Kurt Wallnau***

Program Manager ***Lorraine Martin***

(signatures on File)

CDRL: B001
25 March 1994

Data Reference: STARS-VC-B001/004/00
INFORMAL TECHNICAL REPORT
Market Study

ABSTRACT

The Market Study supplies the Central Archive for Reusable Defense Software (CARDS) program with information regarding the current state-of-the-practice of software development and maintenance within the military services. Results of analysis of collected data reflect current practices, as well as identified needs within each military service for establishment of a suitable infrastructure to enable reuse. Full comprehension of the software needs of potential reusers is required before CARDS can address its ultimate goal of facilitating widespread software reuse throughout the DoD. The focus of results has therefore been geared toward facilitating the CARDS process of institutionalizing software reuse within the Department of Defense (DoD) by more finely concentrating our development, technology transfer and franchising efforts toward satisfaction of identified requirements.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	25 March 1994	Informal Technical

4. TITLE AND SUBTITLE

Market Study
CARDS

5. FUNDING NUMBERS

F19628-93-C-0130

6. AUTHOR(S)

Lynn G. Meheran

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Unisys Corporation
12010 Sunrise Valley Drive
Reston, VA 22091

8. PERFORMING ORGANIZATION REPORT NUMBER

STARS-VC-B001/004/00

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Air Force
Headquarters Electronic Systems Center
Hanscom AFB, MA 01731-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

B001

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution "A"

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

See Abstract Page

14. SUBJECT TERMS

15. NUMBER OF PAGES
68

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

Table of Contents

1 INTRODUCTION

The Market Study was instituted to collect data regarding current DoD software design, development, maintenance, and technical support activities. Military services' software activities (development and maintenance) were surveyed, and the following candidate data were gathered: characteristics of software produced or maintained; contracting and technical processes; and hardware platforms and operating systems. Ultimately, this study is designed to ensure that CARDS principles can be applied throughout the DoD, and to ascertain whether our development, technology transfer and franchising efforts are properly focused to support existing development efforts.

Data from collected results have been converted into usable form to help support CARDS franchising activities. Market Study results may therefore be used to help define the type of information requested from franchise participants or target organizations to gauge their readiness to pursue reuse. Additionally, analysis of collected data may facilitate franchise efforts, by providing lessons learned that may have applicability to the franchising efforts.

This deliverable is the final version of this series ([1], [2], [3]) of working documents. It reflects all data received to date, final results of responses to the abbreviated questionnaire and applicable analyses that may offer benefit to various facets of the CARDS program.

2 MARKET STUDY STRATEGY

The following critical steps constitute the strategy for implementing the Market Study. Initially, a test case was conducted at Electronic Systems Center, Hanscom Air Force Base, to evaluate our current approach, and results were used to refine the methodology for widespread usage.

Contact Electronic Systems Center Reuse Project Officers (ESC RPOs) to test initial interview questions and approach

Conduct initial test case to validate interview questions as well as questionnaires, and refine market study strategy

Devise market study strategy for each service, based upon their typical development approach

Solicit support and assistance from Office of the Secretary of Defense (OSD) and software/reuse experts in each service to reach the appropriate people

Target major commands in Air Force, Army and Navy

Refine market study strategy for each service

Collect data, analyze results and provide report to Government

Gather data, perform initial analysis

Refine strategy, interview questions and questionnaires

Develop abbreviated questionnaire to facilitate large-scale data collection

Institute data collection on large scale

Integrate market study with CARDS Organizational Analysis for Reuse (COAR)
Complete mid-term data collection
Perform in-depth analysis of results
Interleave results of initial COAR into market study results
Write up results of market study
Review results with representatives from each service and DoD
Deliver interim report to Government

Complete long-term data collection
Perform in-depth analysis of results
Write up results of market study
Review results with representatives from each service and DoD
Deliver final report to Government

3 INTEGRATED TARGET SURVEY AUDIENCE

Department of Defense military service, civilian employees and direct support contractors will be targeted to yield sufficient information to enable the CARDS program to determine the state-of-the-practice in software development and maintenance. Those individuals specifically targeted are the following:

High-level business managers (e.g., representatives of Program Executive Officers (PEOs), Designated Acquisition Commanders (DACs), as well as Product Center managers)

Lower-level business managers directing/overseeing development/maintenance (e.g., System Program Office (SPO) level representatives)

Lower-level technical managers directing/overseeing development/maintenance

Engineers/Developers

4 MAIN OBJECTIVES OF MARKET STUDY

The following objectives are intended to provide a context for the Market Study, and to define the focus of current efforts.

4.1 Conduct Survey

Determine how systems development and maintenance are presently being conducted

Explore reuse knowledge held by managers and subordinates

4.2 Conduct Analyses of Survey Results

Ascertain potential barriers to the implementation and institutionalization of software reuse and causes

Determine fit of CARDS products with organizations' and military services' needs

Determine how CARDS can best assist organizations in accomplishing their reuse missions

Identify potential synergies between CARDS and surveyed organizations and military services

5 QUESTIONNAIRE/INTERVIEW QUESTIONS

Reuse Project Officers (RPOs) at Electronic Systems Center provided the initial target interview subjects. ESC RPOs were tasked by their two-letter organizations to provide input toward development of the ESC Software Reuse Implementation Plan. This plan resulted from an Air Force Materiel Command request for each major command to actively respond to the DoD Software Reuse Vision and Strategy's goal of instituting widespread reuse. This version of the questionnaire and set of interview questions have been retailored to delete ESC-specific references, thereby enabling future use of this material throughout the DoD.

5.1 Engineers'/Developers' Questionnaire

A questionnaire for engineers and developers has been created to facilitate self-administration, as well as for use during personal interviewing. Since we anticipate dissemination to many persons in each organization, self-administration offers a reasonable alternative to exclusive use of full-scale interviews. Nevertheless, selective interviews of this sample group may be conducted.

Note to questionnaire respondents: When responding to a list of possible answers, please check only one item, except where alternate instructions are provided.

Personnel Information

Capture characteristics and experience of personnel developing and maintaining software.

What is your title or position, and grade or rank?

How long have you been doing software development?

What educational degrees have you achieved?

Are you pursuing any advanced degrees? In which field(s)?

What application areas have you worked in? Please describe your experience in these application areas.

Have you moved around a lot from one application area to another? __ yes __ no.
Have you specialized in a particular application area? __ yes __ no.
If yes, do you believe that you know a great deal about those specific applications? __ yes __ no.
In which application areas do you consider yourself an "expert"?

Within which application area (domain) do your development efforts lie? __ Communications __ Surveillance/Sensors __ Command and Control __Satellite Communications __ Intelligence __ Other (please specify).

Software Reuse Exposure/Experience

Determine how much general exposure to reuse has been achieved within the services; help refine current and future approaches.

Are you familiar with software reuse (its principles, possible goals, supporting technology)? Please indicate your opinion of your extent of exposure to reuse on a scale of 1 to 10, with 10 representing full exposure and knowledge, 1 representing no exposure or knowledge, and 5 representing moderate exposure.

1		5		10
none	little	moderate	more	full

How did you gain your knowledge of reuse?

Have you ever heard the use of the term "domain" to signify an application area? __ yes __ no.

Are you familiar with the concept of domain analysis? __ yes __ no. If not, skip next question.

To what extent do you believe domain analysis can help you facilitate the reuse of existing components from legacy systems, and help you identify existing

commonalities between analogous systems? Please indicate your assessment on a scale of 1 to 10, with 10 representing full assistance, 1 representing no assistance, and 5 representing moderate assistance.

1		5		10
none	little	moderate	more	full

Are you familiar with the concepts of domain engineering and domain modelling? __ yes __ no.

What is your understanding of the purpose and scope of a software architecture?

Are you aware of major programs, such as Software Technology for Adaptable, Reliable Systems (STARS), Central Archive for Reusable Defense Software (CARDS) and Portable, Reusable, Integrated Software Modules (PRISM), in which software reuse plays a significant role? __ yes __ no.

To your knowledge, are there any reuse advocates (champions) or reuse experts in your organization of whom you can ask reuse-related questions? __ yes __ no
Who are they?

Has software reuse been considered for any program you have worked on? __ yes __ no.
If yes, at what point was reuse considered: __ up front, therefore impacting the entire development process, or __ considered only for maintenance and Post-Deployment Software Support (PDSS)?
Was reuse pursued with __ components originally designed for future reuse, were components __ reengineered to make them suitable for reuse, or __ other (please explain)?

If you have been involved in a program in which reusable components were used, please answer the following questions (if you haven't reused components, please skip to question (TBD):

What type(s) of component was reused? Please check all that apply:
__ domain model __ requirements __ architecture __ designs __ specs
__ code __ test suites __ documentation __ other (please specify).

Were the components being reused originally developed for future reuse? __ yes __ no.

Where did the components originate?

Where did you get them from?

Had the components undergone any qualification or certification process?
__ yes __ no __ don't know.

Did you encounter significant problems in: __ using as is, __ modifying, __reengineering or __ integrating these components? Please place check next to each that is applicable. Please specify the nature of the difficulties.

Were any problems encountered that fall outside these areas? __ yes __ no.
If so, please specify.

Who reused these components, a __ contractor or __ Government team?

What was your impression of the quality and reliability of components being reused, and why?

How would you describe your software reuse experience(s):
__ extremely positive __ somewhat positive __ neither positive nor negative
__ somewhat negative __ extremely negative.
Please explain the rationale for your feelings.

In your opinion, was delivery of software __ accelerated or __ delayed due to reuse of existing components?
If your answer was yes to either possibility, why do you believe this occurred?

If you have worked on a program making use of reusable components, or developing components for future reuse, would you have done anything differently? __ yes __ no. Do you have any recommendations that could improve the process for your organization or command?

Have you ever received any education or training related to software reuse? __ yes __ no.
If yes, where did you receive this education/training?
__ provided on the job __ part of undergraduate course work __ part of graduate course work.
Please provide details about college/university courses or seminars/tutorials attended.

Would you be interested in taking a course in software reuse? __ yes __ no. If so, what would you expect to gain from this course?

Do you have any difficulty in being approved for requested training (e.g., does workload preclude your availability, is funding unavailable)? __ yes __ no. If so, please explain.

What sort of technical support do you believe would be necessary if your organization were told that they would practice software reuse? Is there anything else that your organization should do to make the change process easier for everyone to deal with?

Attitudes About Software Reuse

Capture their feelings about reuse, to help services define potential problems they might encounter.

What is your reaction when the word reuse is associated with software development?

Do you have any preconceived ideas or assumptions about software reuse, either positive or negative? __ yes __ no. If yes, please explain.

Do you have any reason to avoid reusing:
components developed by internal organizations? __ yes __ no.
components developed by external organizations? __ yes __ no.
If your response was yes to either possibility, please specify your reasons.

Do you believe that barriers to the successful practice of software reuse exist within your organization? __ yes __ no. If your response was yes, please explain.

Would you like to learn more about software reuse? __ yes __ no.

Current Development Approaches

Determine how software development and maintenance are currently conducted; provide context to help services define direction for future reuse initiatives.

What is the mission of your organization?

What system(s) are you currently developing?

How is your application area (domain) organized to handle software development?
What layers of management exist, and what are the responsibilities of each layer?
How many managers have control over your work?
Ask for functional organization chart.

How much of your current development effort is contracted out?

Has the amount of contracted-out development changed during the past five years?
__yes __ no __ don't know.

How is systems engineering work accomplished, by __ in-house experts, by __ outside consultants, or __both?
If in-house engineers are used, are they involved in the resulting development program?

What languages are being used in current development and maintenance?
__ Ada __ C __ C++ __ FORTRAN __ Other (please specify).

What hardware and operating system are being used in current development efforts?
Has software been ported to different hardware? __ yes __ no. If so, from which platform to which?
Has attempted porting to different hardware __ succeeded or __ failed?

What software development process(es) is being used by your organization?
__ 2167A __ waterfall __ spiral __ rapid prototyping __ other (please specify).
Have you worked under more than one process, in your current organization or in outside organizations? __ yes __ no. Please specify all processes that apply, and those organizations where they were practiced.
In your opinion, are all these processes well-defined, formalized, understood by all, and adequately documented? __ yes __ no.
Can they be measured? __ yes __ no. If yes, are they being measured? __ yes __ no.

How does management, in your opinion, manage technical risk on all programs on which you have worked? Please check only one for each program.
Do they actively plan to prevent problems from occurring? __ yes __ no.
Do they only "firefight" when problems arise? __ yes __ no.
Do they practice some combination of planning and firefighting? __ yes __ no.

To your knowledge, are any cost analysis or prediction models used to assess risk, and/or analyze the impact of software reuse on projected programs? __ yes __ no.
If so, what models are used?

Even if no models are currently used, are you aware of any particular models?
__ yes __ no.
If so, what are these models?

Metrics

Determine types and extent of measurement currently in practice within the services.

What types of metrics are collected, __ AFR 800-43 - Software Management Indicators, __ CECOM STEP metrics, __ Software Engineering Institute metrics, or __ other (please specify), and for what purpose?
Are metrics designed to evaluate the effectiveness and efficiency of completed software collected? __ yes __ no.
Are prediction metrics collected to determine what can be expected of developed software? __ yes __ no.

Are software reuse metrics collected? __ yes __ no.
If so, what specific metrics are collected, and for what purpose?

Do you have access to statistically validated metrics? __ yes __ no.

If you answered yes to the previous question, what kinds of validated metrics are being collected?

Have you ever evaluated metrics to offer suggestions for refinement? __ yes __ no.
If so, were those metrics related to __ software and/or __ software reuse?

Do you use any tools that support metrics collection (e.g., AdaMat for collection and analysis of software reuse metrics)? __ yes __ no. What tools are being used? Do these tools support analysis as well as data collection? __ yes __ no.

5.2 Interview Questions for Management

A set of interview questions has been developed for all three categories of managers to be canvassed. Many managerial personnel will be personally interviewed, since the current set of questions is far too extensive for self-administration, and necessitates on-the-spot tailoring, dependent upon each person's responses to specific questions. Nevertheless, as data collection continues, thereby enabling the tailoring of sets of interview questions, managerial personnel may ultimately be provided with questionnaires for self-administration. Critically important questions in the initial set of interview questions will be highlighted, and the interviewer will ensure that these questions are answered. The remaining questions will be tailored toward the experience and knowledge level of the person being interviewed. The attached questions are representative of those that will be used to interview management.

Note to survey questioners: When responding to a list of possible answers, please check only one item, except where alternate instructions are provided.

Organizational/Personnel Information

Capture characteristics of organizations in which software development and maintenance are being conducted; better understand the context in which each manager operates.

What is your role and mission in this organization? What are your responsibilities? Do they differ from your position as reflected on the organization chart?

How long have you held your current position? What was your previous position and responsibilities?

Within which application area (domain) do your development efforts lie? __ Communications __ Surveillance/Sensors __ Command and Control __Satellite Communications __ Intelligence __ Other (please specify).

Who is involved in your organization's software development or maintenance efforts? __ Federally-Funded Research and Development Center (FFRDC) __ military/civilian engineers __ contractor __ support contractor.

How do you determine the success of your software efforts?
How do you measure the performance of your subordinates?

What is the experience level of your employees working in the application areas for which you are responsible? __average, __high or __low.
What is the mix of experienced and inexperienced employees working on these applications?
On average, how many years have they been devoted to these applications?
How many related projects have they been working on?
How many of your employees do you consider to be experts in this class of applications?
What is the education level of your employees?
Have any employees received training in software reuse? __ yes __ no.
If so, what was the nature of the training?
Do you feel that this training was sufficient? __ yes __ no. If not, where was it lacking?
Would you be willing to commit resources to enable your employees to receive training? __ yes __ no.

If you discovered a deficiency in terms of the application experience and expertise of your personnel, are resources available to enable you to hire consultants or experts?

Software Reuse Exposure/Knowledge

Determine how much general exposure to reuse has been achieved within the services; help refine current and future approaches.

How much exposure to software reuse have you had? Are you familiar with its principles, concepts, possible goals, terminology, supporting technology? Please indicate your opinion of your extent of exposure on a scale of 1 to 10, with 10 representing full exposure and knowledge, 1 representing no exposure or knowledge, and 5 representing moderate exposure.

1		5		10
none	little	moderate	more	full

How much exposure to software reuse have your employees had?
Please indicate what percentage has been exposed, how they became familiar with reuse, and how thorough is the extent of their knowledge.

Are you aware of major programs, such as Software Technology for Adaptable, Reliable Systems (STARS), Central Archive for Reusable Defense Software (CARDS) and Portable, Reusable, Integrated Software Modules (PRISM), in which software reuse plays a significant role? __ yes __ no.

Do you believe that Ada can bring specific benefits to a software reuse program? __ yes __ no. If so, which benefits? (Look for response of encapsulation or information hiding.) If not, why do you believe Ada is being promulgated?

Have you ever heard the use of the term "domain" to signify an application area? __ yes __ no.
If so, can you identify the domain in which your applications fit?

Are you familiar with the concept of domain analysis? __ yes __ no. If so, what does this mean to you?

On a scale of 1 to 10, to what extent do you believe domain analysis can help you facilitate the reuse of existing components from legacy systems, and help you identify existing commonalities between analogous systems? 10 represents full assistance, 1 represents no assistance, and 5 represents moderate assistance.

1		5		10
none	little	moderate	more	full

Are you familiar with the concepts of domain engineering and domain modelling? __ yes __ no.

What is your understanding of the purpose and scope of a software architecture?

Would you be interested in attending and/or having your employees attend a short course or training in reuse? __ yes __ no.
If so, what would you hope to get out of this training?
What would you like the training to focus on?

Software Reuse Experience

Explore in depth management's experiences in reusing components or in developing reusable components.

Are you __ actively practicing software reuse, and/or __have you practiced software reuse in the past? If yes, are you involved in any organizations that promote reuse? __ yes __ no. If yes, what are they?
What were the goals of past reuse programs?
What are the goals of your current reuse program? Have these changed significantly due to lessons learned from past efforts? __ yes __ no.
Is software reuse being pursued on an __ individual, __ team or __organizational level?
Are reusable components created by an independent development team? __ yes __ no. If yes, how many people are involved and what are their roles?
How did you go about identifying software reuse opportunities? Were you assisted through some __ formal or __ informal means?
Did you prototype a system using reusable components? __ yes __ no.
Have you collected data on the up-front investment required to practice software reuse? __ yes __ no.
What did you find was required?
Are these data disseminated beyond your organization? __ yes __ no.
If so, to whom?

Have you ever incorporated reusable components into any program that you have managed? __ yes __ no.
What were your objectives in pursuing reuse of available components?
Were requirements from users __ sufficiently flexible to enable you to do reuse, or were they __ so rigid that this posed a significant problem to your pursuit of software reuse?
Did you perform any prototyping of the system at any point during the life cycle, either to __ help refine requirements or __ test the applicability of potential reusable components?
Did you trade off requirements in order to reuse components? __ yes __ no.

When requirements were being examined, was software reuse a prime consideration? __ yes __ no.
Will reuse be considered during Post-Deployment Software Support (PDSS)? __ yes __ no.
Was software reuse used as an evaluation factor during source selection? __ yes _ no.
Did regulatory guidance __ hinder or __ help your reuse efforts?
Were any software reuse experts available to help guide your effort? __ yes __ no.
Did your contracts people use any novel wording/approaches to address any unusual issues that might be peculiar to software reuse? __ yes __ no.
Were typical proposals modified to enable you to effectively evaluate technical and management reuse approaches? __ yes __ no.
If so, what were these modifications?
Do you have any results that would indicate their usefulness? __ yes __ no.
What either facilitated or impeded the reuse of available components?
Did you change the process by which you evaluate project performance to include reuse? __ yes __ no. If yes, what changes were made?
Did you have to negotiate any license or maintenance agreements? __ yes __ no.
Did you consider the possibility of business practices and technical risk to your program from software reuse? __ yes __ no.
If so, how did you assess, evaluate and consider potential technical and business practices risks?
How did you act on the results of your investigation?
Were your actions successful? __ yes __ no. Why/why not?
What type(s) of component was reused? Please check all that apply:
__ domain model __ requirements __ architecture __ designs __ specs
__ code __ test suites __ documentation __ other (please specify).
How did you identify potential components for reuse?
Were the components being reused originally developed for future reuse? __ yes __ no. If yes, did they still require major modification for your program to use them? __ yes __ no.
Was there a set of minimum criteria used to evaluate potential reusable components? __ yes __ no.
How did you evaluate these components to determine their applicability to your program?
Were they evaluated against established criteria? __ yes __ no.
Were they subjected to __ rigorous internal reviews, or did you use __ another means of evaluation? Please specify.
How did you determine the potential reusability of available components?
What was your impression of the quality and reliability of components being reused, and why?
Were the components reused as part of a __ test bed or __ pilot project? __ no.
Were Commercial-Off-The-Shelf (COTS) and/or Government-Off-The-Shelf (GOTS) components available? __ yes __ no.
If so, were they evaluated for potential use? __ yes __ no.

Where did the components originate? Who developed them?
How were they acquired? Was a process used that differed from the usual? __ yes __ no. If yes, how did it differ?
Where did you get them from?
If you obtained the components from a reuse library, which library(ies) was accessed?
Had the components undergone any qualification or certification process? __ yes __ no __ don't know.
What classification methodology was used (e.g., faceted classification, semantic net)?
Was it easy to (check all that apply):
__ navigate through the library,
__ determine which components were potential candidates,
__ examine these components,
__ extract selected components,
__ work with the library to resolve any licensing issues.
Did you encounter any problems in using the reuse library? __ yes __ no. If so, please explain.
Have you provided usage reports back to the library? __ yes __ no.
Were the components __ used as is, or __ customized for your application(s)? If customized, were they __ modified, or __ reengineered? How significant were the required changes?
Did you encounter significant problems in: __ using as is, __ modifying, __reengineering or __ integrating these components? Please place check next to each that is applicable. Please specify the nature of the difficulties.
Were any problems encountered that fall outside these areas? __ yes __ no. If so, please specify.
Was integration complicated by reuse of components? __ yes __ no. If so, how?
Was integration testing more thorough due to incorporation of reusable components? __ yes __ no.
Have any testing practices changed since introducing software reuse? __ yes __ no.
Do you have formal agreements that govern use of these components? __ yes __ no.
If so, with which organization(s), and what is the nature of each agreement?
During which portion of the life cycle were these components incorporated?
Was software reuse considered for maintenance phases, as well as original development? __ yes __ no.
Were potential components used to rapidly prototype a system? __ yes __ no.
Did you use Value Engineering Change Proposals (VECPs) to motivate your contractors? __ yes __ no.
How would you describe your software reuse experience(s):
__ extremely positive __ somewhat positive __ neither positive nor negative
__ somewhat negative __ extremely negative. Please explain the rationale for your feelings.

In your opinion, was delivery of software __ accelerated or __ delayed due to reuse of existing components? If your answer was yes to either possibility, why do you believe this occurred?

Do you consider your reuse of existing components to have been successful? __ yes __ no.

If yes, do you have any estimates of the time or money saved through software reuse? __ yes __ no.

Was productivity enhanced? __ yes __ no. Do you consider the enhancement to have been significant? __ yes __ no.

If no, why do you believe that this software reuse effort failed?

What was the reaction of your employees to software reuse?

If the reaction was negative, do you think that advance training in reuse or technical guidance documentation would have been helpful? (If currently considering reuse, mention possibility of arranging for presentation of Systems/Software Engineer's course material or access to Engineer's Handbook.)

Did you have to "sell" the idea of software reuse to them? __ yes __ no.

Was a particular segment of your development team less receptive to software reuse? __ yes __ no. If so, could you characterize these persons (e.g., less or more experienced, younger, older)?

Do you believe that incentives are necessary to convince employees to practice software reuse? __ yes __ no. If so, what incentives do you believe would be effective?

Have you attempted to use any of these incentives? __ yes __ no.

How successful were your attempts?

If successful, how did you go about validating success? Did you base your conclusions on the __ success of reuse, or did you __ canvas your subordinate managers or employees?

Were lessons learned fed back to refine incentives? __ yes __ no.

Based upon your software reuse experience, what do you feel must happen or work out right for reuse to succeed? Where would you hate to see something go wrong?

Have you ever managed the development of software components for future reuse? __ yes __ no.

Was this development __ dictated, or did you __ actively pursue it?

Where did the funding come from to support the additional work required to develop reusable components?

On a scale of 1 to 10 in terms of additional levels of complexity and difficulty (10 is greatest complexity and difficulty), how did development of reusable components differ from that of development for no planned reuse?

Do you have any estimates about how much more it cost to develop software for future reuse? __ yes __ no. Please provide us with your estimate.

Did you use any cost models to estimate the financial impact of this development on your program? __ yes __ no. If so, please specify which one was used.

Did you develop these components as part of a reuse pilot project? __ yes __ no.

Did you use Computer-Aided Software Engineering (CASE) tools at any point during the life cycle (with emphasis on those tools supporting software reuse)? __ yes __ no.

Which tools were used, and for what purpose?

Did you feel that their use was valuable? __ yes __ no.

Did you use any special reuse technology in developing these components? __ yes __ no.

If yes, how did you become aware that this software reuse technology was available? Did this technology provide the level of assistance that was originally anticipated? __ yes __ no.

If you didn't use any special technology, are you aware of any that could have facilitated the development process? __ yes __ no.

Have you any thoughts about the best means of ensuring that other developers are aware of the product(s) you created, and/or the reuse technology that facilitated development, and have access to them? __ yes __ no.

Do you have any lessons learned that could benefit the remainder of your organization's development community? __ yes __ no.

What benefits do you believe accrued to your organization from pursuit of software reuse?

Were these components developed against any standards or guidelines regarding completeness, quality and applicability (form/fit/function)? __ yes __ no.

If so, what standards or guidelines were used?

Is the software thoroughly documented? __ yes __ no. If not, why not?

Did you place your developed components in a reuse library? __ yes __ no.

If so, which one?

Is this an __ in-house or __ external library?

Who is responsible for its management?

Will your organization maintain this software? __ yes __ no.

If not, do you know how version control and documentation will be managed? __ yes __ no.

If another organization will be maintaining the software, do you believe that it will be difficult for them to maintain? __ yes __ no. Why/why not?

Have you established any methodology to govern the evolution of this software to maintain its reusability? __ yes __ no.

If you have no defined methodology, do you have any ideas about how you could maintain its reusability? __ yes __ no.

Did you collect any metrics on the development process or the resulting component(s)? __ yes __ no.

Have you any evidence that this developed software is being reused? __ yes __ no.

If not being reused, could these components be refined, so that they could gain widespread usage? __ yes __ no.

Have you received any feedback from reusers to help you gauge the effectiveness, efficiency, quality, validity and reliability of your developed software? __ yes __ no.

Is there anything that you would do differently next time when developing reusable components? __ yes __ no. If so, please present your ideas.

Are you aware of any products resulting from this development effort that could be altered from specific to general, and reused by a wider user base? __ yes __ no.

Have you ever practiced "black box" reuse, in which only the interfaces with other components are of concern, rather than the inner workings of the component? __ yes __ no.

If your organization is not reusing software, why not? __ no assets __ too costly __ legal problems __ contractor resistance __ other (please explain).

Attitudes About Software Reuse

Capture their feelings about reuse, to help services define potential problems they might encounter; explore assistance that could be provided by reuse community.

How do you feel about software reuse? If reaction is negative, ask whether success stories would help convince him/her of potential value of software reuse.

Do you believe that software reuse is a reasonable solution to the service's need for greater productivity in the face of downsizing? __ yes __ no.

Do you have any ideas about alternative solutions? __ yes __ no. If so, please specify.

How do you feel the institution of software reuse would impact your organization?

Are you considering using reusable components in an impending acquisition or developing reusable components? __ yes __ no. Would you welcome guidance? __ yes __ no. If so, would you consider using the CARDS Direction Level or Acquisition Handbook as a guide to recommended approaches? __ yes __ no.

What services could the software reuse community provide to best assist you in minimizing the impact of change on your organization while meeting the objectives of software reuse?

What do you need to implement software reuse?

Is there a minimum toolset, document set, etc. that is required? __ yes __ no. (Should perhaps be used as last question to those who haven't instituted reuse.)

Ownership/Rights/Incentives

Determine whether managers, at all levels within an organization, are aware of pertinent ownership, rights and incentive issues; capture ideas and recommendations on these issues by exploring each person's thoughts.

Are you familiar with ownership and rights issues as they pertain to different software components? __ yes __ no.

What types of components do you think the Government should own?
__ domain models __ requirements __ architectures __ designs
__ specifications __ code __ test suites __ documentation
__ other (please specify).

How would you make an ownership decision? What criteria would you apply?

Do you believe that certain components should be owned only under certain circumstances? __ yes __ no. If so, what are the components in question, and what circumstances pertain?

How would you define the benefits, liabilities and responsibilities associated with ownership?

What level of rights should be retained by the contractor, to maximize benefits to your organization and its contractors?

How is your software development or maintenance effort being contracted?
__ firm fixed price __ fixed price + incentive fee __ cost plus
__ cost reimbursement __ other (please specify).

Would you be willing to __test the validity of ownership/rights decisions, or to __test contractual language and contract vehicles in any upcoming program? __ no.

How do you feel about providing contractors with incentives to incorporate software reuse into projected programs and to develop reusable components?

Do you have any ideas about how contractors should be incentivized to invest more in order to ensure the success of software reuse initiatives? __ yes __ no. (Need to mention in which ways they will be expected to invest.)

Have you tested your contractors' receptiveness toward specific incentives, or their use in specific contracts? __ yes __ no.

How did you go about this, what were the applicable incentives, and how successful were the incentives?

Exploration of Ideas Regarding Software Reuse

Determine potential for reuse within an organization; capture ideas about means of approaching the institution of reuse.

If the mission of your command were modified to incorporate software reuse, and you were tasked to implement reuse within your organization and throughout your programs:

How would you prioritize issues in your organization relative to software reuse?

What steps would you take initially, to pave the way for reuse? __ pilot project(s) __ resolve barriers __ other (please explain).

What would be your next step?

Would you __ implement in progressive steps, or __ mandate? Why?

What sort of technical support would be necessary?

In your class of applications or domain, do you believe that there is potential for software reuse? __ yes __ no.

Are there other systems being developed within your command that bear similarities to yours? __ yes __ no.

Are you aware of any generally common interfaces between types of components in this domain? __ yes __ no.

Do you have a feel for any percentage of commonality between systems in this domain? __ yes __ no.

Is this domain stable and mature? __ yes __ no.

Do you believe there are opportunities for strategic or short-term partnerships for development? __ yes __ no.

What are the critical characteristics (e.g., component class - spreadsheet, module size, operating environment) of the architecture that pertain to your class of applications?

In your opinion, are any of these characteristics common to all applications? __ yes __ no. If so, which are common?

What do you consider to be critical general characteristics of components that you would be asked to reuse?

Do you have any additional requirements for specific types of reusable components? __ yes __ no.

What do you consider to be critical components of a command-level software reuse infrastructure that would facilitate reuse of existing components or development of reusable components? Reuse infrastructure can be defined as a combination of policies, processes, technology and personnel required in an organization to incorporate reuse into the software development process (Franchise Plan).

Do you feel that there are holes in the existing infrastructure? __ yes __ no.
Where are the holes?
What do you believe should be done to ensure the success of software reuse initiatives?

Cultural Factors

Identify potential impediments to widespread institutionalization of systematic software reuse, and specific organizational context.

Do you believe that any impediments to software reuse exist:

__ within your organization,

__ within your command/service,

__ within DoD.

Please specify any technical, business or personal (applicable either to yourself or your employees) barriers that you perceive.

What do you believe should be done to reduce their impact or eliminate them altogether?

Why do you believe that these barriers exist? Are there some organizational constraints that are responsible? __ yes __ no. If so, what are they?

Do you believe that your personnel will encounter barriers or perceive that barriers exist? __ yes __ no.

Are these barriers due to lack of knowledge about software reuse? __ yes __ no.

Are they inherent to the practice of software engineering? __ yes __ no.

If yes to either question, what do you believe are the reasons for their existence or any pertinent misconceptions?

Do you have any ideas about how to neutralize their impact? __ yes __ no.

Have you heard other managers or your employees express any negative opinions about software reuse? __ yes __ no. If so, please explain the nature of the comments.

Based on past experience, do you believe that your employees are flexible and adaptable to changing circumstances? __ yes __ no.

Can you characterize those who appear to you to be most resistant to change (e.g., are they younger, older, less or more experienced, etc.)?

**** Don't repeat here if addressed under reuse of components ****

Do you believe that incentives are necessary to convince employees to practice software reuse? __ yes __ no. If so, what incentives do you believe would be effective?

Have you attempted to use any of these incentives? __ yes __ no.

How successful were your attempts?

If successful, how did you go about validating success? Did you base your conclusions on the __ success of reuse, or did you __ canvas your subordinate managers or employees?

Were lessons learned fed back to refine incentives? __ yes __ no.

** End of repeat **

Are your developers matrixed to other organizations or managers? __ yes __ no.

If so, do you believe any serious (or potentially serious) communication difficulties exist? __ yes __ no.

Mission/Current Software Development Practices

Determine how software development and maintenance are currently conducted; provide context to help services define direction for future reuse initiatives; identify issues that have potential of undermining software reuse.

What is your view of your command's mission regarding software development and maintenance?

Has there been a change in the mission due to the institution of software reuse? __ yes __ no. If applicable, how would you prioritize each of these objectives?

Are you aware of any existing impediments that interfere with accomplishment of your mission? __ yes __ no.

Do you have any suggestions to offer that may eliminate these impediments or at least lessen their impact? __ yes __ no.

Could software reuse facilitate this process? __ yes __ no.

What has historically been the focus of development within your organization - __ high-level components such as architectures and designs, or production of __ lower-level components, like code, or __ both?

Are any cost analysis or prediction models used to assess risk, and/or analyze the impact of software reuse on projected programs? __ yes __ no.

If so, what models are used?

Even if no models are currently used, are you aware of any particular models? __ yes __ no. If so, what are these models?

How thoroughly documented are the systems produced by your organization?

In your opinion, is documentation sufficient to enable totally independent organizations to easily and fully understand all components for the purposes of maintenance? __ yes __ no.

Are all systems __equally documented, or is there __variation within your organization?

What are your most pressing concerns (personal as well as professional) relative to current software development practices:
__ high costs __ low productivity __ late delivery __ other (please explain).

What are the greatest difficulties you have encountered in accomplishing your development or maintenance mission? Why do you believe that these difficulties exist?

Do any __ service-level organizations direct or in any other way impact your current practices (to include software development, maintenance, acquisition, contracting), or do influences only occur at the __ command level?

Is there any means of information exchange established at command level to identify valuable reusable components or supporting technology, as well as sharing experience with components/technology, metrics, lessons learned about use of various development processes and usage of various acquisition and contractual approaches? __ yes __ no.
If not, would you like to see such an exchange established? __ yes __ no.
Would you use such information? __ yes __ no.
Would you support such an exchange by actively providing information? __ yes __ no.

**** Deleted if engineers/developers in organization are also being canvassed ****

What languages are being used in current development and maintenance?
__ Ada __ C __ C++ __ FORTRAN __ Other (please specify).

What hardware and operating system are being used in current development efforts?
Has software been ported to different hardware? __ yes __ no. If so, from which platform to which?
Has attempted porting to different hardware __ succeeded or __ failed?

**** End of deletion ****

What are projections for hardware and operating system use for the next five years?
Are there plans in place to transition current efforts to different hardware? __ yes __ no.

What software development process(es) is being used by your team?
__ 2167A __ waterfall __ spiral __ rapid prototyping __ other (please specify). Why was this approach selected?
Have you ever used a different process? __ yes __ no.
If so, why did you make the decision to change the process?

Do you have any lessons learned from use of either process that could benefit others in your command? __ yes __ no.
In your opinion, are all such processes in your command well-defined, formalized, understood by all, and adequately documented? __ yes __ no.
Do you believe that this engineering process can be qualified and measured? __ yes __ no. If yes, are process metrics collected? __ yes __ no. What metrics are collected?
Are any automated software life cycle tools part of your software development process? __ yes __ no. If so, which are being used?
Could include: __ requirements analysis __ design __ development __ configuration management __ testing __ documentation development __ maintenance.

What, if any, process improvement and/or quality control management techniques or mechanisms are being used?
Do any standardized policies or procedures apply? __ yes __ no.

What technology and software engineering environments are you and your team (including contractors) currently using in developing software?
Would you like to use additional technologies? __ yes __ no.
Why do you want to use these technologies?
What (or who) is preventing you from using these technologies?
How do you hear about available technology?
How do you go about assessing available technology?
Do you have a feel for your projected technology needs? __ yes __ no.
Can you specify anticipated needs? __ yes __ no.
Would you like to improve the way in which technology is used in your programs (e.g., increase its use, while decreasing personnel resources required)? __ yes __ no.
Do you have any lessons learned tied to use of specific technology that could benefit other managers at your command? __ yes __ no.

Current Command-Level Software Reuse Efforts

Determine extent of software reuse currently underway, and assess the command's ability to institute successful software reuse programs.

Are you aware of any software reuse activities taking place at a parallel level to your organization within your command?

Do you believe that the highest management levels support the institution of software reuse in your command? __ yes __ no.
If no, why? What is or is not being done that leads you to this conclusion?

If yes, why? What actions are being taken by management to advocate reuse and support its implementation (e.g., __training, __investments in personnel, __ investments in technology, __ other)?

Are you aware of any efforts being considered or being implemented to systematize software reuse (i.e., to ensure creation of a level playing field for Government developers and contractors involved in reuse)? __ yes __ no.

Where do you think your command is in terms of its capability to implement software reuse? What do you believe needs more work?

Tools/Technology Usage

Determine whether the services are actively acquiring or supporting the development of appropriate technology.

Are you supporting (either financial or technical) the development of technology designed to assist your organization in __ developing software, __ reusing software, or in __ developing reusable software?

If so, what types of technology are you supporting, and how?

Do they facilitate __ original development, __ development of reusable assets and/or __ reuse of existing assets?

Could developed tools or technology be refined, so that they could gain widespread usage? __ yes __ no.

Have you established a technology plan? __ yes __ no.

If so, what is it designed to accomplish?

Metrics

Determine types and extent of measurement currently in practice within the services.

What types of metrics are collected, __ AFR 800-43 - Software Management Indicators, __ CECOM STEP metrics, __ Software Engineering Institute metrics, or __ other (please specify)?

Are metrics collected to enable you to better manage your team? __ yes __ no.

Which metrics are collected?

Do you believe that they're effective in helping you manage? __ yes __ no.

Are metrics designed to evaluate the effectiveness, efficiency and quality of completed software collected? __ yes __ no.

Are prediction metrics collected to determine what can be expected of developed software? __ yes __ no.

Are results fed back to improve the development process? __ yes __ no.

Are software reuse metrics collected? __ yes __ no.
If so, what specific metrics are collected?

Do you have access to statistically validated metrics? __ yes __ no.
If yes, what kinds of validated metrics are being collected?

Have you ever evaluated metrics to offer suggestions for refinement? __ yes __ no.
If so, were those metrics related to __ software and/or __ software reuse?

Do you use any tools that support metrics collection (e.g., AdaMAT for collection and analysis of software reuse metrics)? __ yes __ no.
What tools are being used?
Do these tools support analysis as well as data collection? __ yes __ no.

5.3 Library Management

This set of questions is designed to capture important library information, should any particular organization have developed an extensive component library. This library would have to either offer access to other organizations or represent an unusual approach beyond that of a Program Support Library that performs mainly configuration management functions. These data would be used to maintain current information on libraries accessible to others, and monitor other libraries with differing methodologies. The questions would be asked of library operations people.

Who is responsible for management of this library?

How long has the library been operational?

Have you established criteria and procedures to satisfy security and integrity requirements? __ yes __ no.

What types of components are stored in your library?

Have you developed criteria to validate these components for new applications? __ yes __ no.

What cataloguing or classification scheme is being used?
Does this conform to any established standards (in your service/within the DoD)? __ yes __ no.

How is qualification and/or certification of components undertaken?
How many levels of certification are possible?

What are the criteria associated with each level of certification (i.e., what criteria must a component meet in order to attain each level of certification)?

Are components qualified, certified and validated against domain requirements? __ yes __ no.

What library mechanism is being used for browsing, extraction, etc.?

Do you have a published set of standards and policies that govern library operation? __ yes __ no.

What types of support services do you provide?
Have they been evaluated to determine their usefulness to library users? __ yes __ no.
Have you canvassed your user base to determine their level of satisfaction with offered services? __ yes __ no.

Do you offer education and training for your users? __ yes __ no.
If yes, what kind of training is offered?
If yes, what kind of education is offered?
What are applicable policies and procedures relative to training and education (e.g., who can attend, when, is there a cap on course cost, etc.)?

Against which criteria are potential users evaluated for the purposes of library access control?

Does your library interoperate with any others? __ yes __ no. If so, which libraries?

Once components have been selected for placement in the library, are they evaluated in terms of their usefulness to users? __ yes __ no.
Is there any specific component or tool that has proven to be particularly popular or useful? __ yes __ no. If so, which tool and/or component?

What library usage metrics are collected?
Has library usage changed over time? __ yes __ no. If so, how has usage changed?

Are metrics collected to gauge the effectiveness of the library itself? __ yes __ no.
If yes, what specific metrics are collected?
Have these metrics been statistically validated? __ yes __ no.

Are users later queried to determine whether the components they extracted were actually used in an application? __ yes __ no.

Has your library evolved over time, based upon results from collected metrics and lessons learned? __ yes __ no. If so, what is the nature of the evolution?

Do you have agreements to govern the relationships between suppliers, subscribers and the library? __ yes __ no.

What liabilities do you think that you face in operating your library?

6 ESC RPO INTERVIEWS

Electronic Systems Center was tasked with developing a command-level software reuse plan by Headquarters, AFMC. Consequently, the command asked each two-letter organization to appoint a Reuse Project Officer (RPO) to represent the two-letter in development of the reuse plan, and perhaps ultimately to represent ESC and the Air Force in instituting reuse within each two-letter organization. Since ESC is the sponsor organization for CARDS, it was decided that this organization would serve as our first test case for the purposes of conducting interviews and collecting data. The ESC RPOs available for our interviews varied widely in terms of their experience and knowledge, both about current development and maintenance activities and software reuse. Some are solely developers, while others occupy managerial positions or perform a dual role of management and development. Because the characteristics of the RPOs were not consistent, we weren't able to use one questionnaire, and the sample didn't coincide with our original plans for data collection. Due to the inconsistent breadth of expertise and knowledge among the RPOs, coupled with the overview we gained of multiple organizations, rather than in-depth investigation of a particular organization, we weren't able to fully validate our present questionnaire and set of interview questions. We did provide tailored questionnaires to two interviewees, though, and if they are completed and returned, we will use this specific data to continue the validation process.

Although we were unable to thoroughly assess the usefulness of all of our existing survey material, we were able to gauge the value of many specific questions which were used in most interview situations. For instance, the following questions were asked of most RPOs:

How much exposure to software reuse have you had?

To your knowledge, are there any reuse advocates, champions or reuse experts in your organization of whom you can ask reuse-related questions? To whom would you turn if instructed to actively pursue reuse?

Would you have done anything differently (from your perspective of having worked on a program using or developing reusable components)?

Do you have any recommendations that could improve the process for your organization or ESC?

Do you believe that barriers to the successful practice of software reuse exist within your organization?

What languages are being used in current development?

What hardware and operating system are being used in current development efforts? We also asked questions about porting, depending upon responses received to previous questions.

What software development process is being used by your organization?

Are metrics being collected, what types are collected, and are there any tools being used to support metrics collection and analysis?

Do you believe that incentives are necessary to convince employees to practice software reuse?

What services could the software reuse community provide to best assist you in minimizing the impact of change on your organization, while meeting the objectives of software reuse?

We also explored issues relating to terminology, such as the use of the terms domain and domain analysis, and queried them about their views regarding software reuse.

We were also able to answer some specific questions for the ESC community, including current hardware and language usage. For instance, hardware currently being used ranges from a variety of platforms supporting UNIX, to VAX 4300s, 6610s and 6620s, HP 720s, Macs, Sun SparcStations, Motorola hardware, AlphaWorks, 486-based PCs, a RISC-based processor and a Hughes product, AMD44. Several applications currently resident on VAXs are being ported to client/server environments. Some organizations are also exploring a move toward open systems, in part to decrease existing dependence upon specific types of hardware. COTS/GOTS (such as utilities, databases, communications) and CASE usage (e.g., design tools) is not typical, but appeared to be well-received when in use. Languages in use include Ada, C, C++, FORTRAN, Pascal and assembler.

Depending upon the level of exposure to reuse stated by each RPO, as well as each person's general experience and expertise, questions were selected that were judged to be best suited to each individual. Consequently, interview results are not consistent between respondents, and summarizations are therefore difficult to make. Nevertheless, we did gain useful data for ESC as well as the CARDS program. This section of the document will present ESC-specific information, as well as conclusions that the market study team drew from these initial interviews, with particular emphasis placed on those results with potential application to CARDS efforts.

These results are designed to enable the ESC reuse community to gain insight into the readiness and willingness of its developers and maintainers to practice reuse. These data will also provide information regarding the RPOs' opinions about those steps they would like to

see instituted at ESC to help facilitate reuse. Please keep in mind that the results presented here may not be characteristic of the organizations that the RPOs represent, and therefore should not be generalized. They were personal opinions expressed by one individual within each organization, and therefore may reflect nothing more than each person's ideas. Nevertheless, this does not diminish the importance of carefully considering these opinions for their possible application to the challenge of instituting widespread reuse throughout the ESC community.

Due to the wide diversity regarding knowledge level, current development and maintenance practices, reuse practices, etc., this information in many cases is only partially applicable to the CARDS effort. Expectations of obtaining usable data were greater than the results obtained, given the broad command-level, rather than organizational perspective gained from discussions with the RPOs. One cannot simply convert basic data gained through the interview process to information of direct benefit to CARDS. The information gleaned from such interviews may sometimes serve only to enrich our knowledge base. Nevertheless, it is always useful to gain others' perspectives on development, maintenance and reuse, and future data collection efforts may further corroborate those results that reflect general agreement among the RPOs. Finally, experience gained in conducting interviews will ultimately benefit the CARDS Organizational Analysis for Reuse task, by facilitating the team preparation process.

6.1 General Comments

This section will demonstrate the tremendous variety of experience represented by the ESC RPOs. Those RPOs who are currently unschooled in software reuse still regard reuse as involving only code. Although reuse is not a novel concept to most RPOs, its practical application is. Despite this fact, several RPOs had become well-acquainted with existing reuse issues that the larger reuse community is currently grappling with, in part through their tasking as RPOs. For instance, one RPO mentioned concerns about ownership, liability, certification, who will maintain the components, incentives to get personnel to even search for potential reusable software, the pressing need for standards, little guidance provided by superiors and cultural barriers that must gradually be resolved.

Despite some RPOs' lack of detailed knowledge about reuse, they nevertheless raised some issues that reflect concern about infrastructure issues, assessment of reuse claims, and the way in which reuse will be instituted by ESC officials. Several stated flatly that reuse should not be mandated as Ada was, because it will probably suffer the same failure as the Ada mandate, and merely increase resistance on the part of ESC personnel. Some also mentioned that they consider perceived risk and inadequacy of documentation to constitute major barriers to reuse at ESC. As one individual put it, we need to maintain the pedigree of documentation with the software. Other concerns centered around assessment issues, such as how to appraise capabilities and incurred risks with complex code - in other words, ESC personnel are looking for some means of quantifying risks, in addition to determining whether code functionality satisfies its claims.

Infrastructure issues included the following: who is going to pay for development of reusable software, who will maintain it, and who will pay for the maintenance; and, who will be in

charge of the databases of reuse information at ESC. Other concerns mentioned by the RPOs included the volatility of requirements baselines, and the difficulty in challenging claims about reusable software, and worst case, discovering that these claims are unsubstantiated. Finally, some mentioned concern about the acceleration of integration problems with reusable components, when these components were not originally designed to be integrated into other packages.

6.2 Results and Conclusions

The material within this section is grouped into categories to facilitate review, and more easily identify areas of concern. Many of the general issues raised by the Reuse Project Officers (RPOs) were not surprising to the CARDS interviewers; for the most part, these concerns are familiar to the reuse community. However, some ESC-specific issues and proposed solutions are particularly novel, and warrant close inspection by the CARDS team.

The interviewers were able to obtain frank and sometimes controversial comments from ESC RPOs by ensuring that their input would remain confidential and their privacy protected. Consequently, RPOs and their organizations are not specifically named, and wording has been altered in such a manner as to protect the intent of the statement, while cloaking specific wording that might identify particular individuals. It is absolutely essential that we continue to protect the confidentiality of our subjects; therefore, there is only a small core of persons who has access to the full range of interview results.

6.2.1 Availability of Information about Reuse and Communication Issues

There was general agreement among the RPOs that insufficient information is currently widely available at ESC about reuse in general (including typical terminology), ESC or other DoD success stories, and explanations of the anticipated impact of the institution of reuse on personnel at Hanscom Air Force Base. Although they agreed that increased communication is necessary within the ESC community to introduce personnel to the concepts and principles central to successful domain-specific reuse, they differed regarding which communication media would offer the greatest likelihood of success. Many felt that the Hansconian (base newspaper) would be a good place to start, and others believe that some sort of on-site demonstration of current capability (perhaps within the CARDS or PRISM programs) could best reach ESC participants by stimulating their interest. To be most effective, these on-site demonstrations should be rotated among several large buildings to reach the largest possible audience. The CARDS team recognizes, however, that such presentations could have a significant impact on personnel allocation, since current demonstrations require manpower for operation. Other means of communicating information wouldn't drain resources, however, and might be viable options for ESC. RPOs recommended using libraries or bulletin boards to make such data widely accessible, and some also recommended that additional forums (e.g., conferences or industry-wide forums) for exchange of information be devised.

Interviewees expressed the conviction that communication is the key issue in facilitating successful integration of reuse at Hanscom. Once reuse preliminaries have been presented, they would welcome more detailed information about the DoD and Air Force reuse initiatives, their impact on the development of the ESC Software Reuse Implementation Plan and future

bearing on reuse initiatives currently underway at ESC. Most were unaware of current reuse activities at Hanscom, including programs in which reuse is an integral part, as well as significant tasking from the DoD and Air Force to institute reuse. Many were also generally unaware of the directive nature of these reuse initiatives.

The RPOs would also like to fully comprehend the rationale for the reuse community's contention that reuse will succeed and deliver anticipated benefits, since many believe that the potential of reuse has yet to be proven. For example, one RPO stated that his finance people have informed him that reuse will work only if the developer is familiar with the software to be reused; otherwise, development costs will be comparable to development from scratch. The RPOs identified the following objectives for an ESC education program: expose personnel to reuse; get out there with information - make reuse and its successes visible; inform personnel about available services; and prove to personnel that reuse will work

One RPO recommended that once small successes are achieved, ESC should ensure that these successes become more visible, perhaps by publicizing results and ensuring that this information is widely disseminated, so that the learning process gradually filters through the ESC organization. Although this could offer an ideal means of gradually instituting reuse, we believe that such an approach may not be practical at ESC, because of the critical necessity for rapid change due to tight budget constraints and the need for increased productivity in the face of diminished resources. Nevertheless, one cannot question the value of publishing positive results, if they are not classified, company or Government proprietary or otherwise protected, thereby making this information available to ESC personnel. This raises another issue, though - how to motivate employees to read, accept and assimilate such material. Prior prejudices against such information will necessarily affect its influence; however, since so many persons are presently unaware of reuse and therefore have no preconceived judgements, this is an ideal time in which to reach ESC personnel.

Communication was also cited as a critical link between organizations, especially where reuse is concerned. When active information exchanges occur, organizations can capitalize upon opportunities to benefit from internal reuse. Additionally, communications loops can facilitate the dissemination of success stories to the wider ESC community, thereby educating personnel about current activities, and hopefully lessening potential barriers to more widespread reuse institution. Unfortunately, our data indicate that active pursuit of information gathering at ESC is rare, and few commonality studies are underway. Nevertheless, there is an exception to this rule - one two-letter chief is actively canvassing other organizations, as well as programs within his own, to identify potential reuse opportunities.

Although the RPOs indicated the crucial importance of educating ESC personnel about reuse, one RPO strongly expressed the conviction that we need to first ensure that personnel are educated about software fundamentals. Without requisite training in metrics, estimating, scheduling and testing, for example, schooling in reuse will not be effectively implemented. In other words, a sufficient base of knowledge and experience in effective software engineering or development must be present before the potential benefits of reuse will be fully realized. Consequently, another RPO emphasized the critical importance of ensuring that all

technical personnel receive training necessary to ensure their effectiveness.

CARDS training courses were explored as potential education and communication vehicles. The RPOs agreed that the CARDS Introduction to Reuse course would be a helpful and welcome tool to educate ESC personnel about reuse, and most expressed a personal interest in attending this course. (Please note: this course was subsequently provided to RPOs, and was well received.) Once reuse concepts and principles are presented and assimilated, then the CARDS Applications Engineering course would have a sufficient audience to justify its presentation at Hanscom. However, our results indicated that few persons could fully benefit from this more advanced course at the present time, due to the present lack of a foundation of knowledge about reuse.

In conclusion, our interviews have indicated that enhanced communication, in a variety of forms, could help lay the groundwork for acceptance and openness to the notion of widespread reuse by ESC personnel.

6.2.2 Perceptions About Reuse

The RPOs expressed a general acknowledgement of the necessity for reuse due to budgetary and resource constraints, despite some of their experiences with or knowledge about past reuse which have not been altogether positive. One RPO believes that reuse was possible in one instance only because it was contemporary software - older software that is more closely tied to particular hardware impairs another's ability to use it. Consequently, some RPOs feel that one of the major goals of a reuse program should be to move toward open systems. Others believe that problems that have arisen with GFE components (e.g., greater functionality promised than actually delivered) have led to a predilection to distrust reusable components. As one RPO put it, "Reuse isn't the issue - quality of the reusable software is". Another stated that reusable software is only as good as the person who developed it. Another problematic issue that is related to a component's future reusability is the inadequacy of documentation to support its use. Several persons cited the delayed documentation of many systems at ESC, a potential stumbling block in instituting reuse.

6.2.3 Experiences with Reuse

Many RPOs are aware of successful reuse, either personally experienced or through colleagues. Almost exclusively, these examples of successful reuse were instances of a contractor reusing its own previously developed code to develop current systems. Consequently, one RPO believes that reuse is extremely difficult if Government personnel are not kept up-to-date on industry's accomplishments, so that they are aware of potential reuse opportunities. Another commented that he would prefer an industry-wide forum to a central repository as an information source, because he believes that repositories filter information, and their perspectives may not be his.

Despite these RPOs' positive experiences, common major difficulties have been encountered. These center around the following: determining whether percentage of reuse claims made by competing contractors are likely; how best to estimate the possibility of successful completion of a specific level of reuse; what is the possibility of error (as well as how significant an error);

and, how to determine which components can be reused as is, and which require major reengineering. Consequently, some believe that there needs to be more rigor in how source lines of code estimates are made, and how best to evaluate them. These concerns may also provide a focus for the reuse community, since these existing difficulties were voiced by many RPOs, and may typify potential problems encountered during development efforts with reusable components.

Causation of these difficulties may be tied to the acquisition system itself, which in the opinion of several RPOs encourages overly optimistic proposals; in other words, contractors bid far more reuse than is realistic to remain competitive. The only incentive for contractors to reuse available components is to propose favorable costs and schedules when bidding on a contract. Further compounding these problems is the fact that some high-level management "supports" reuse on the surface, but doesn't comprehend the risks nor fully understand software. This situation makes it particularly difficult to explain how a program could be behind schedule and over budget when integrating reuse, which is supposed to provide time and budget savings. The sum total of RPO comments led the team to conclude that widespread education is necessary to dispel any misconceptions about reuse, and ensure that all players are operating on a level playing field in terms of knowledge and expertise.

6.2.4 RPO Suggestions to Facilitate Development and Reuse

The material presented in the following paragraphs addresses many facets of the life cycle; therefore, the presentation of ideas follows life cycle development, and the order is not meant to signify any interpretation of relative importance.

6.2.4.1 Prior to Contract Award

One RPO believes that reuse should be addressed very early in the inception of programs, during Acquisition Strategy Panels. This person's idea was to have "Generals" require that presenters explain reuse plans or provide rationale for not pursuing reuse. This person reasoned that since "Generals" establish the format and the material to be presented, why not insert reuse considerations at this juncture if they are fully committed to reuse, to further the objective of reuse institutionalization?

6.2.4.2 Acquisition and Development

Once a development program has been formulated, another RPO recommends that industry develop system specifications, to achieve the long-term goal of maximizing the potential for reuse. This person believes that industry experts are best equipped to recognize and fully exploit parallels between previous work they have accomplished in the field and proposed new development.

Once acquisition is underway, several RPOs believe that internal incentives should be used to motivate Government personnel to work toward achieving reuse, at least initially. However, once success in reuse is achieved, it will be self-perpetuating.

In order to create an environment favorable for the pursuit of reuse at ESC, another idea was to create a Reuse SPO to fund reusable component development efforts, thereby reducing the risk for individual programs and reducing resistance to undertaking such work.

Another pointed out that it might make sense to develop and implement programs on separate platforms. Since hardware is inexpensive, and this approach could enable two teams to develop in parallel, the Air Force might have a program delivered in a more timely fashion. This concept could also enhance reusability, since the two teams would have had to develop with interfaces in mind, which should reduce the integration impact when these components are reused.

Extensive oversight is currently difficult, due to the lack of manpower. Consequently, one RPO suggested that if ESC has a good working relationship with a contractor, they should provide only limited Program Office oversight. Additionally, one RPO pointed out that SPO staff need to be located where development is taking place, so that they can establish a relationship with the contractor that will enable them to function as part of the team. Another cited the current inconsistent use of metrics to closely monitor programs. Each of these comments reflects organizational viewpoints that differ considerably. This heterogeneity underscores the unique nature of individual organizations and the programs they support, issues which we must assess and continually monitor in our franchising efforts, to ensure that our solutions are best matched to the particular organization's needs.

6.2.4.3 Production/Deployment and Maintenance

One RPO believes that contractors, rather than the Government, should perform system maintenance, because of their familiarity with the system. It is this person's conviction that this will result in lowered total life cycle costs.

6.2.5 Reuse Community Assistance

When asked how the reuse community could assist in implementing reuse, RPOs recommended the following: supporting education efforts; providing standardization; ensuring that well-commented code is available for reuse; and minimizing the integration impact of reusable code.

6.2.6 Conclusions for Consideration by the CARDS Teams

6.2.6.1 Training Issues

Training was particularly emphasized by several RPOs, and for a variety of reasons. One person cited the importance of training those persons in responsible technical positions who are not viewed as computer literate. Another emphasized the critical importance of ensuring that all technical personnel receive a solid foundation in core technical disciplines; such training is viewed as the minimum required to provide sufficient groundwork to secure a reasonable chance of success for reuse initiatives.

Many of the RPOs expressed interest in the CARDS Introduction to Reuse course. However, many cautioned that the two-letters should be specifically targeted for involvement, since they will direct much of the future reuse activity at ESC, and must fully understand pertinent issues and ramifications for their organizations if reuse is to succeed. The RPOs also expressed opinions regarding training focus and course material for presentation, and supplied several suggestions for our training team. First, they believe that presentation of concrete examples

of programs' pursuit of reuse would facilitate comprehension, and dispel the myth that reuse entails only code and documentation usage. One person offered the suggestion that the course include a description of the process by which the PRISM program developed the Generic Command Center. They also recommended that a formal briefing of the CARDS and PRISM programs be included as part of the one-day course at ESC. Finally, one person stated that novices must fully understand the process required to successfully transfer software to other programs, and that presentation of this material should be part of the introductory course.

A final recommendation by one RPO, and subsequently embraced by others, was that a "Reuse Road Show" could offer another effective means of reaching the ESC community with reuse information. This "road show" concept would incorporate use of workstations to demonstrate products or capabilities at a central site with maximum exposure. These demonstrations would periodically be relocated to reach the widest possible audience. In conjunction with these exhibits, published information, as well as handouts and brochures would be available to participants.

The following material is a quick synopsis and reiteration of points made by the RPOs that may influence current CARDS training efforts, and provide some worthwhile ideas for future updates to course material. For more detail, please refer back to the text.

- General reuse information (e.g., terminology) is not readily available.

- RPOs identified the following objectives for an ESC education program: expose personnel to reuse; get out there with information - make reuse and its successes visible; inform personnel about available services; and prove to personnel that reuse will work.

- RPOs agreed that the Introduction to Reuse course would be useful, and most are interested in attending. Once they have a sufficient grounding in reuse, then the Applications Engineering course would be helpful.

6.2.6.2 Success Stories

Several RPOs emphasized the importance of effectively reaching the wider ESC development and maintenance community, especially the decision-making two-letter chiefs, about the tremendous potential of reuse. One suggested approach is to support one or two two-letters to ensure success of their reuse efforts, and then publicize the results of these accomplishments, thereby grabbing the attention of other two-letters. Another RPO emphasized the importance of our identifying such "windows of opportunity", and capitalizing upon them by employing our expertise to ensure successful implementation of reuse in particular programs.

6.2.6.3 Reuse Promised vs. Delivered

One criticism of reuse efforts is that they often don't deliver the amount of reused components that is originally bid. Due to greatly overestimating what can actually be reused, and underestimating what proportion must be reengineered, programs experience significant schedule and financial impact. Because of personal experiences with these situations, some RPOs are convinced that for software to be reused, it must have originally been designed and developed for that purpose. If not so designed and developed, then reuse would in actuality become reengineering, which is expensive and time-consuming. One RPO also believes that

this pertains to prototype systems as well, because associated documentation will likely be minimal and not representative of the system as it exists.

Another difficulty centers around contractually addressing the system's requirements regarding performance for those components that the Government wants to have reused. For instance, some products advertised as compatible with others fail to define expected performance when systems are linked together, and threshold performance limits may be exceeded. This is another undefined area in which contractual wording must properly reflect the Government's desire to balance reuse goals with satisfaction of tight performance constraints (a dilemma not easily solved, but nevertheless highly critical). It also highlights the critical necessity of resolving remaining legal issues that may constitute impediments to successful reuse.

6.2.6.4 Documentation and Related Issues

Many existing components are not documented. Although the official policy is that no software will be released without documentation, the reality is that it often is, and critical supporting documentation often considerably lags behind build releases. Unfortunately, DoD documentation specifications don't specify what documentation is essential and appropriate for reusable software. Although pertinent description of such documents may be difficult, especially when compounded by the formalities of contractual language, additional details need to be specified, and the contractor must fully understand what is required and what material must be explained. One person believes that reuse documentation must include a "how to" reuse manual, as well as expected requirements, design and test information. Otherwise, reverse engineering will be necessary to determine how to interface reusable software with newly developed components. Nevertheless, the integration issue will become increasingly difficult in the future, as modules are developed for very specific purposes, and may have been designed to operate as stand-alones, not part of a larger package of functionality. After all, it is extremely tough to build components to meet other than immediate requirements, as it may be extremely difficult to fully anticipate future requirements.

One RPO expressed concern about the lack of available COTS and GOTS products that have been fully tested in applications where their use was required for success. This RPO's experience indicated that Government-developed tools aren't generally useful, because although contractors are amenable to developing these tools, no one is willing to undertake long-term support. He was encouraged to hear that the CARDS program is actively using COTS/GOTS products and demonstrating their use.

6.2.6.5 Ownership/Rights

Whenever legal issues arose during our discussions, RPOs expressed concern about resolution. In fact, one RPO stated that one of the important tasks (and one probable means of facilitating reuse) that we could undertake was to solve these legal problems, since two-letters cannot resolve these issues independently. Contractors are understandably reluctant to place anything in reuse libraries, due to uncertainty about protection of copyrights and patents, unauthorized usage, and collection of appropriate fees or royalties.

In addressing general legal issues, one RPO stated that legal questions should be presented only to contracts people, rather than technical personnel, since the technical side's only concerns are who owns it and how much it costs.

Some RPOs would like to remove barriers to enable contractors to own more components; however, one person believes that such willingness would suggest a degree of open-mindedness that simply does not exist. In his words, they prefer to keep the enemy they have and know.

Ownership issues also arose when one organization undertook usage of available components. Although code was lifted, supporting documentation was not readily available. Consequently, the top-level design suffered, the development team couldn't proceed, and the schedule suffered delays.

Finally, some RPOs recommend that award fees or some other form of incentive fee be made available, so that some mechanism is available to stimulate contractors to search for and find the best solutions.

6.2.6.6 Program Control/Domains/Product Lines

Some RPOs supported our current efforts to promulgate domain management and control. One RPO's experience led him to conclude that the potential for productive reuse vastly increases when planning and execution of reuse takes place within a single organization with cognizant responsibility for a particular domain. He is convinced that this common organization must be present for reuse to succeed. He also emphasized that components need to be associated with common architectures, since much less reuse takes place when components are general purpose and slated for use across multiple projects.

This RPO also stated that major consolidation is taking place within the DoD, which raises the potential for significant reuse. However, he contends that this consolidation must be monitored to track those organizations actively pursuing reuse, determine what each is attempting to do and the scope of their activity, and finally identify good candidates for future reuse. He envisions this monitoring and assessment task as one which should be undertaken by the reuse community. Another recommended service that could be provided by the reuse community is development of common specifications or standardized requirements, or at least the means of expressing them. If this were available, then one could predict what data would be supplied by contractors.

Several RPOs were unfamiliar with the term "domain", and preferred use of "product line" in its place. The term "domain" was viewed as a further complication of an already complex issue, and the RPO simply didn't want to get involved.

A final issue which affects ESC's efforts to institute reuse centers around control. Several RPOs stated that ESC can't manage the reuse effort at the Headquarters (ESC/CC) level, through issuance of directives, because one can't control reuse without directly controlling the programs; however, the two-letters can control reuse, since they do have control authority over programs, and each roughly constitutes a domain. One person viewed the ESC/CC and

reuse community's role as providing support to two-letters, as well as assistance in accomplishing their mission, rather than control over their reuse efforts. This person expressed the opinion that ESC/CC needs to work closely with program authorities responsible for reuse. Since the two-letters are the decision makers that control the domains, they will ultimately control the magnitude of reuse at ESC. Consequently, ESC must try to influence them to take advantage of what we in the reuse community have to offer; in other words, we must make reuse worthwhile for them, so that they are motivated to reuse, rather than being forced into it.

7 NAVAL UNDERSEA WARFARE CENTER (NUWC) VISIT RESULTS

Part of the Market Study team visited the Naval Undersea Warfare Center Division, Newport, RI (NUWCDIVNPT, RI). The team interviewed a group that supports the Technical Direction Activity (TDA) task for Program Offices responsible for command and control systems on submarines. NUWC has traditionally acquired software exhibiting limited design for reuse for its sonar, targeting and control of weaponry. The group with whom we spoke is developing a Reuse Process Report to guide future acquisition efforts away from specific purpose processors and languages toward more commercial COTS products, the use of Ada, and transition to an open systems standard interface architecture. Their goal is to use COTS wherever possible in the general-purpose processing portions of command and control systems. They are also moving toward Next Generation Computer Resources (NGCR) standards that include open systems and COTS, use of standard backplanes, operating systems, and networks, and use of standard database access, such as Structured Query Language (SQL). Their initial intent is to focus on how they want to pursue reuse, rather than grossly populating an uncertified set of library components.

NUWC is actively searching out possible reuse opportunities in existing systems that may have applicability to the New Attack Submarine (NAS) development. The NAS development plans include evaluation of potential donor systems and identification of segments that could be used in the NAS. Selected subsystem Program Offices will be funded to modify and hand off the existing segments to the NAS system integrators. Each subsystem Program Office will be responsible for reuse within its portion of the overall system. Ultimately, NAS will make use of (i.e., reuse) many existing subsystems. However, the predominant concern is not reuse, but rather, integration issues. Integration will follow a platform approach with each major subsystem being managed as a separate program. All subsystems will be integrated onto one platform.

Several technical and management issues complicate the pursuit of such large-scale reuse: 1) different display hardware used in various subsystems; 2) architectures and technology specific to subsystems will be integrated with possibly different architectures and technology present in other subsystems; 3) multiple timing issues will require synchronization; 4) many layers of architecture may exist; and 5) visibility into white-box components in previous systems will give way to lack of visibility into the new black-box components (e.g., COTS).

Design for reuse represents a change from past development practices. Consequently, its institution may result in resistance. Additionally, organizational issues have not been fully

explored. Optimizing the application of designing for reuse can be applied at the NAS system platform infrastructure level.

The interviewees offered several observations about reuse. They believe that little reusable code will be available from legacy systems until such systems are designed for reuse. Since development Program Offices are concerned about meeting deadlines, and partitioning between development and maintenance exists, the current acquisition structure impedes reusable component availability. A reusability test for code is required - code should pass a quality as well as a functional test. The existence of object-oriented design is providing a better mechanism to pursue reuse, through the encapsulation of data and functionality within an object.

The interviewees also expressed concerns about obtaining the information necessary to pursue many elements of reuse. For instance, available documentation typically doesn't adequately describe the rationale for specific design decisions. In addition, little or no assistance is available to help one analyze code to determine its potential reusability, and identify the detailed information that is necessary to capture the essential elements for domain analysis. A methodology for production of good software is in place, but the difficulty they have encountered while performing domain analysis is obtaining the appropriate information necessary to accomplish the task. System level support must be present to provide resources for domain analysis.

They have explored many different methodologies for conducting a domain analysis, and have found Firesmith (a notation developed by Donald Firesmith of ASTS; product name is ADM-3; product documented in "Object-Oriented Requirements Analysis and Logical Design: A Software Engineering Approach") to be most suitable for their purposes, since it offers a systems approach to software definition. One of the difficulties encountered in analyzing potential methodologies is the lack of a standard notation. Objectmaker (developed by Mark V Systems, 16400 Ventura Blvd., Encino, CA 91436) has the benefit of offering nearly every possible notation, but doesn't adequately document the rules for each. However, in one person's opinion, Objectmaker offers an impressive reengineering capability, and another plus is its ability to support the reverse engineering of code to a design. The designer can get a snapshot of the design captured in a segment of code, and thus effectively abstract the design from the code.

Not surprisingly, the quality of code is of primary importance to this group, and they are working to define it as part of their methodology. They have designated certain criteria that quality code must meet, including use of encapsulation, good software engineering, and limited dependencies.

This group at NUWC has spent some time defining the methodology to facilitate reuse, but has yet to create the products that will support library definition and the resources to populate a repository. They are currently in a process definition phase, and are committed to promulgating the value of following a formalized process or approach, in the hopes that it will be adopted as a departmental or NUWC objective. Since the ability to consistently capture knowledge and develop high quality code does not currently exist, and the software

engineering process is too ad hoc, they believe that provision of some control over that process and the inclusion of a formal method offers a viable improvement option for NUWC, and perhaps eventually for the wider Navy community as well.

One interviewee had some ideas about the best means of encouraging the reuse of products. First, the products themselves must be reusable, and programmers must have a certain level of confidence in their use. This confidence in the code must flow from confidence in the software qualification procedures. However, he would caution that quality for reuse does not necessarily imply appropriateness for a particular application. For example, one can't put absolute timing requirements on a generic solution that may be available in a reuse library; although the component may be a good generic, its performance can't be guaranteed for specific applications.

Secondly, since reuse requires extensive knowledge (such as where and how to get components), when this knowledge becomes available to programmers, then reuse will happen. However, an incentive- driven benefits system must be established to encourage good programmers to reuse and develop for future reuse, to ensure that current skeptical mindsets are effectively overcome. Thirdly, use of good software engineering, or a good object-oriented approach, rather than a particular language, will tip the balance toward reuse, since disciplined engineering can make reuse happen, regardless of which language is used. One can have good software engineering without reuse, but cannot have reuse without good software engineering. Fourthly, sponsorship and/or managerial support is required with financing and extracting information from systems application personnel, and may also provide unexpected benefits and help resolve additional issues. Fifthly, the way in which Program Offices are set up provides no incentive to pursue reuse, since specific programs are entities in themselves. Success is determined by acceptance test and determination of whether the tactical system meets requirements, not against productivity or quality measurements. Until the software system generation process is overhauled to address these issues, widespread reuse will not occur.

Those interviewed stated that the reuse community could provide assistance in the following areas: 1) help define what reusability is; 2) define what quality of code for reuse really means; 3) determine critical thresholds for values to help interpret measurements used in determining reusability; 4) sell the DoD bureaucracy on reuse, to encourage the establishment of reuse as an objective for SES-level positions; and 5) provide a potential list of appropriate incentives for use with contractors.

One interviewee also had some suggestions for possible consideration by management personnel to facilitate reuse. These suggestions included: 1) establishing a software reuse infrastructure; 2) making the decision to have one combat system, thereby narrowing the breadth of the domain; 3) making reuse part of personal objectives, to provide the impetus for change and widespread institution of reuse, and encourage formal information exchanges between systems and software personnel; 4) providing management support, both programmatic and financial, as well as funding to ensure the success of reuse initiatives; and 5) helping to define appropriate engineering practices for reuse.

8 SURVEY QUESTIONNAIRE

An abbreviated version of the Market Study questionnaire was developed to capture a core set of critical questions. This questionnaire was disseminated on a widespread basis to Government employees, both military and civilian, in an effort to gain a considerable amount of development, maintenance and reuse information. The realized benefit to CARDS was the rapid accumulation of large amounts of data at a lower cost than that possible under the original plans.

Following are the cover letter and abbreviated questionnaire that were mailed or e-mailed to our target recipients. We sent out 751 questionnaires and received 121 responses, 66 from the Air Force, 22 from the Army, 18 from the Navy, 2 from NASA, and 13 from miscellaneous organizations. Responses were received from the following organizations within the services: Air Force - SSC, SMC, SM-ALC, SCXX, USSTRATCOM, USAF/7CG, OO-ALC, CSC, USAFA, OC-ALC, AFIT, AFTAC, HQAF, SWSC, WR-ALC, RL, WL, ESC, LXGE, AFGWC, ETAC, CPD/TM, AFSAA, SCSR, CSSQ/SCID, CS/SCP, ASC, CSGP/WCSE, NTF and SAF/AQK; Army - AEC, IRMD, AMSRL, SFAE, TEC, TACOM, AMCRD, STRICOM, AMSEL, USASSDC, ODISC4, USAISSDCL, CSSCS, USACOE, ARDEC, TACMIS, USAOEC; Navy - NARDAC, NAPEP, NUWC, NAWCWPNS, NSPCC, FNOC, NSWCDD, NASC, FMSO, NSWC, NAWC, NWSCC, NRaD, NAVCOMTELCOM, and NCCOSC.

The breakdown of personnel responding to the questionnaire is as follows: management responsibilities (56 - Air Force(36), Army(12), Navy(7), NASA(1)), software development(51 - Air Force(25), Army(10), Navy(14), NASA(2)), software maintenance(35 - Air Force(23), Army(4), Navy(7), NASA(1)) and other(26 - Air Force(16), Army(8), Navy(1), NASA(1)).

8.1 Cover Letter for Survey Questionnaire

Reply to ENS 1 Oct 93
Attn of:

Subject: DoD Software Development and Maintenance Market Study Survey Questionnaire

To: Government personnel involved in software development and maintenance

To assist in determining the state-of-the-practice in DoD software development and maintenance, we are asking for your help in completing the attached questionnaire. Data collection will be managed by the Central Archive for Reusable Defense Software (CARDS) program, which is dedicated to DoD's strategy to transfer technology and techniques for library-aided, domain-specific reuse into mainstream DoD software procurements. Survey results will be used to facilitate the widespread institution of software reuse within the DoD.

We would appreciate your cooperation in completing this questionnaire. Please feel free to include any additional comments you may have in response to specific questions in the questionnaire, or in regards to current practices or software reuse at the end of the questionnaire.

Thank you for your assistance. If you have any questions, need assistance or wish to provide more in-depth responses, please contact Lynn Meheran at meheranl@gw1.hanscom.af.mil, or by phone at (508) 443-9700.

Robert Lencewicz
Program Manager, CARDS
Systems & Software Design Center

1 Attachment
1. DoD Software Development and Maintenance Market Study Survey Questionnaire

AF Survey Authorization Number USAF SCN 93-96

8.2 DoD Software Development and Maintenance Market Study Survey Questionnaire

We welcome any input you can provide; partially completed questionnaires are acceptable. Please complete the requested information that immediately follows, as we will be correlating data to organization, job title, etc. We will protect the privacy of your responses. If there are others in your organization or command who could provide us with valuable input, please submit their names and addresses, since we want to obtain the widest possible view of current practices as possible.

Name _____
Rank/Title _____
Organization _____
Address _____
_____
_____
Telephone: (_____) _____-_____
E-mail _____
Major responsibility(ies)
__ Management __ Software development
__ Software maintenance __ Other (please specify)

If you are working on multiple programs, please provide answers for each program in the following manner:

example - you are working on two programs, in which various languages are being used. Program 1 uses Ada and FORTRAN; Program 2 uses C++. In response to question 1, which languages are being used in current development and maintenance, please answer as follows:

1 Ada __ C 2 C++ 1 FORTRAN __ Other

Then, for all subsequent questions that apply to these programs, please answer in the same way. Thanks for your cooperation.

Please fold the completed questionnaire so that the address on the back of the form is exposed, then staple or tape the form together, and return it to us.

Please place a check next to all responses that are applicable.

Current Development/Maintenance Practices

1. What languages are being used in current development and maintenance?
__ Ada __ C __ C++ __ FORTRAN __ Other (please specify)

2. What hardware and operating system(s) are being used in current development/maintenance efforts? Please specify.

3. What software development process is being used by your organization? __ 2167A __ waterfall __ spiral __ rapid prototyping __ other (please specify)

4. How is your software development or maintenance effort being contracted? __ firm fixed price __ fixed price + incentive fee __ cost plus __ cost reimbursement __ other (please specify)

5. Who is involved in your organization's software development or maintenance efforts?
__ Federally-Funded Research and Development Center (FFRDC) __ military/civilian engineers __ contractor __ support contractor

6a. Are metrics being collected? __ yes __ no
6b. If so, what metrics are being collected?
__ Software Management Indicators
__ CECOM STEP
__ Software Engineering Institute metrics
__ reuse metrics
__ other (please specify)

7. Are any cost analysis, prediction or planning models used to assess and manage software technical risk? __ yes __ no. If yes, please list those model(s) that are being used.

8. What are your most pressing concerns relative to current software development practices?
__ high costs __ low productivity __ late delivery __ other (please explain)

Software Reuse

9. Are you familiar with software reuse? __ yes __ no. If your answer is no, please skip to question 29.

10. Do you believe that software reuse is a reasonable solution to the military services' needs for greater productivity in the face of downsizing? __ yes __ no

11. Do you believe that the highest management levels within your organization support the institution of software reuse? __ yes __ no

12. Do you believe that barriers to the successful practice of software reuse exist within your organization? __ yes __ no

13. To your knowledge, are there any reuse advocates (champions) or reuse experts in your organization of whom you can ask reuse-related questions? __ yes __ no

14. If your organization is not reusing software, why not? __ no assets __ too costly __ legal problems __ contractor resistance __ other (please explain)

15. Have you worked on a program in which reuse was practiced? __ yes __ no. If no, please skip to question 29.

16. What type(s) of component was reused? Please check all that apply: __ domain model __ requirements __ architecture __ designs __ specifications __ code __ test suites __ documentation __ other (please specify)

17. Were Commercial-Off-The-Shelf (COTS) or Government-Off-The-Shelf (GOTS) components used? __ yes __ no

18. Was there a set of minimum criteria used to evaluate potential reusable components? __ yes __ no

19. Was reuse pursued with __ components originally designed for future reuse, were components __ reengineered to make them suitable for reuse, or __ other (please explain)?

20. In your opinion, were the components reliable and of high quality? __ yes __ no

21. Was any prototyping of the system conducted at any point during the life cycle, either to __ help refine requirements or __ test the applicability of potential reusable components? __ no

22. Did you encounter significant problems in:
__ using as is, __ modifying, __ reengineering or __ integrating these components?

23a. Are any cost analysis or prediction models used to analyze the impact of software reuse on projected programs? __ yes __ no
23b. If yes, please specify the model(s) used.

24. Did reuse increase productivity? __ yes __ no

25. How would you describe your software reuse experience(s): __ extremely positive __ somewhat positive __ neither negative nor positive __ somewhat negative __ extremely negative

Recommendations

26. Is there anything you would do differently next time in either reusing available components or developing reusable components?

27. Based upon your software reuse experience, what do you feel must happen or work out right for reuse to succeed? Where would you hate to see something go wrong?

28. What services could the software reuse community provide to best assist you in meeting the objectives of software reuse?

29. Do you have any recommendations that could improve the software development or maintenance process in your organization?

30. Can you recommend anyone to whom we should send this questionnaire?

31a. Do you think that the DoD needs a software advocate and/or reuse advocate? __ yes __ no.
31b. If yes, __ software advocate, __ reuse advocate, or __ both?
31c. From which organization, and at what management level?

Thanks very much for your help. We appreciate your input.

Comments/Remarks

Question Number Comment

8.3 Final Survey Questionnaire Results

8.3.1 Current Development/Maintenance Practices

1. *What languages are being used in current development and maintenance?*

Languages currently in use (including numbers of respondents identifying their use), include the following: Ada(84), C(45), C++(24), FORTRAN(41), CMS-2(5), COBOL(19), Clipper(2), BASIC(2), Atlas(6), other HOLs(1), Assembler(26), Simscript(2), Pascal(2), ADS(1), Progress(1), LISP(1), 4GLs(5), COMMAC (in-house developed macro language - 2), J73(1), Jovial(8), Access(1), PL/1(1), Teleuse(1), DEC's DCL(1), SmallTalk(1), J3B(2), 1750 Assembler(1), FoxPro(1), CMS-2M(2) and SPC(1).

2. What hardware and operating system(s) are being used in current development/maintenance efforts?

Hardware currently in use includes: 486s(4), 386s(3) and 286s(2), Digital Equipment Corporation (DEC) VAXs(31), IBM-compatible PCs(23), Apple Macintoshes(2), Sun Workstations(13), AN/UYK-7(1) and AN/UYK-43(1), UYK-20(1) and UYK-44(1), 1553 Bus(1), tester-specific compilers(1), RISC machines(2) and a variety of platforms manufactured by Hewlett-Packard(10), DEC(8), HP RTE(1), Texas Instruments(1), Honeywell Bull(3), Alsys(1), IBM(9), Pyramid(2), Silicon Graphics(6), AT&T(2), Unisys(7), HP/Apollo(1), Teradyne(1), Data General(1), Verdix(2), Rational(3), Cray(4), Harris(2), Mentor Graphics(2), Amdahl(1), Perkin Elmer(1), Apple(1), CDC(1), Sequent(1) and Sun(28).

Operating systems that apply to these types of hardware include: Solaris(3), IBM MVS(3), VAX/VMS(20), MS-DOS(18), OS/2(2), UNIX(31), HPO(1), IRIX(2), CTOS(1), BTOS(1), Nighthawk(1), CTASC II(1), AIX(5), POSIX(1), HP-UX(2), XD Ada(1), VXWORKS(1), DT111/IV(1), AUX(1) and Ultrix(1).

The following is the breakdown of languages, hardware and operating systems by service/NASA.

Air Force

- languages: Ada(52), C(23), C++(13), FORTRAN(32), Assembler(22), 4GLs(5), COMMAC(2), Pascal(1), J73(1), Jovial(8), Atlas(6), Access(1), PL/1(2), COBOL(16), Teleuse(1), DEC's DCL(1), SmallTalk(1), J3B(2), 1750 Assembler(1), Simscript(1), FoxPro(1), BASIC(2) and CMS-2(1).
- hardware: Unisys(5), RISC(2), PCs(12), Sun(19), Sun Workstations(8), HP/Apollo(1), DEC(3), DEC VAXs(22), Teradyne(1), Honeywell Bull(3), AT&T(1), Data General(1), IBM(8), Verdix(2), Rational(3), Cray(4), Harris(2), Silicon Graphics(5), Hewlett-Packard(3), tester-specific compilers(1), Mentor Graphics(2), Apple(2), Amdahl(1), 286, 386 and 486 PCs(5), Perkin Elmer(1), 1553 Bus(1), Texas Instruments(1), Sequent(1).
- operating systems: MS-DOS(16), Solaris(3), UNIX(24), VMS(20), MVS(5), AIX(4), OS/2(5), IRIX(1), HP-UX(2), XD Ada(1), VXWORKS(2), Mac System7(1), DT111/IV(1), AUX(1), Ultrix(2), SEVMS(1), AIS(1), GCOS(2) and DYNIX(1).

Army

- languages: Ada(18), C(11), C++(7), FORTRAN(5), HOLs(1), Assembler(3), Simscript(1), ADS(1), Progress(1), COBOL(1) and Pascal(1).
- hardware: Sun(1), Sun SparcStations(3), Hewlett-Packard(4), Apple Macintosh(1), DEC(1), DEC VAXs (4), PCs(4), 286, 386 and 486 PCs(3), CDC(1), Alsys(1), Pyramid(1), AT&T(1), Silicon Graphics(1), Unisys(1) and Intel(1).
- operating systems: UNIX(7), IRIX-5(1), SUN/OS(4), HO/OS(1), CTOS(1), BTOS(1), MS-DOS(7), VMS(3), Nighthawk(1), CTASC II(1), Alsys First Ada(1), OS/X(1), Mac System7(1), HP/OS(1).

Navy

- languages: Ada(12), C(9), C++(2), FORTRAN(3), CMS-2(3), LISP(1), COBOL(2), CMS-2M(2), Assembler(1) and SPC(1).
- hardware: Sun(5), Sun Workstations(2), AN/UYK-7(1), AN/UYK-43(1), UYK-20(1), UYK-44(1), DEC VAXs(3), other DEC(4), Unisys(1), IBM-compatible PCs(5), Hewlett-Packard(3), HP Workstation(1).
- operating systems: OS/2(1), MS-DOS(7), AIX(1), UNIX(7), VMS(7), MVS(1), POSIX(1), SunOS(2), UNICOS(1), SHARE(1), UP-UX(1), AegisTactical Executive(1) and Solaris(1).

NASA

- languages: Ada(2), C(2), C++(1), FORTRAN(1).
- hardware: PCs(2), Sun(1), DEC(2), IBM(2), Rational(1), RS6000(1).
- operating systems: MS-DOS(1), UNIX(3), VMS(2), MVS(2).

3. *What software development process is being used by your organization?*

Software development processes currently in use (including frequency of usage) are: 2167(1), 2167A(71), tailored or modified 2167A(4), 2167A combined with another process(3), Software Cleanroom Engineering(3), SM-ALC PDSS(1), combination of all approaches(1), waterfall(38), spiral(25), rapid prototyping(36), object-oriented analysis and design(3), systems analysis and design(1), accelerated development(1), development against Navy standard 1679(1), simulation-based development against 2167A(1), development against Air Force standard 7935(1), hybrid of waterfall, 2167A and spiral(1), and no formal method(2). These results, broken out by service, are as follows:

Air Force

- Waterfall - 28
- Spiral - 13
- Rapid prototyping - 18
- 2167A - 44
- Modified/tailored 2167A - 2
- 2167A combined with other process(es) - 3
- Object-oriented analysis and design - 3
- Air Force standard 7935 - 1
- SM-ALC PDSS - 1
- Combination of all approaches - 1

Army

- Waterfall - 7
- Spiral - 8

- Rapid prototyping - 12
- 2167A - 16
- Accelerated development - 1
- Software Cleanroom Engineering - 1

Navy

- Waterfall - 3
- Spiral - 4
- Rapid prototyping - 6
- 2167A - 11
- 2167 - 1
- Systems analysis and design - 1
- Simulation-based - 1
- Navy standard 1679 - 1

4. *How is your software development or maintenance effort being contracted?*

The contracting method being used in development and maintenance efforts includes: firm fixed price(33), fixed price plus incentive fee(10), cost plus(29), cost plus fixed fee(1), cost reimbursement(14), time and materials(6), not applicable due to in-house development(12), level of effort(4), in-house with contractor support(1), and research(1). A breakdown of contracting method by service/NASA follows.

Air Force

- Firm fixed price - 23
- Fixed price plus incentive fee - 6
- Cost plus - 17
- Cost reimbursement - 6
- Other(23) - in-house development(11); level of effort(3); cost plus fixed fee(1); time and materials(3); research(1); not applicable(4).

Army

- Firm fixed price - 4
- Fixed price plus incentive fee - 3
- Cost plus - 7
- Cost reimbursement - 5
- Other(3) - level of effort; time and materials; task order time and materials.

Navy

- Firm fixed price - 5
- Fixed price plus incentive fee - 1
- Cost plus - 3
- Cost reimbursement - 3
- Other(4) - in-house development; in-house development with contractor support; industrial fund (fee for service); time and materials.

NASA

- Firm fixed price - 1
- Fixed price plus incentive fee - 0
- Cost plus - 2
- Cost reimbursement - 0
- Other(2) - level of effort; in-house development.

5. Who is involved in your organization's software development or maintenance efforts?

	Air Force	Army	Navy	NASA
Federally-Funded Research and Development Center	6	2	2	0
Military/civilian engineers	51	19	18	1
Contractor	38	18	6	0
Support contractor	27	19	13	2

One Air Force respondent included an additional group in the list, that of universities.

6a. Are metrics being collected?

Metrics collection is taking place, although inconsistently in terms of its extent and breadth, and combined results are as follows: 75 persons indicated collection of metrics, while 43 stated that their organizations are not collecting metrics. Despite the greater numbers implying metrics use, the services differ considerably in their metrics activity. 40 Air Force respondents cited collection, while 20 don't collect; Army respondents are more heavily weighted toward collection, with 19 citing collection and 5 citing non-collection; finally, the Navy's results are equally distributed, with 11 each citing collection and non-collection.

6b. If so, what metrics are being collected?

Metrics collection within the services and NASA is comprised of the following types of metrics: Software Management Indicators(38), CECOM STEP metrics(4), Software

Engineering Institute metrics(20), reuse metrics(12), and others(20). A breakdown for types of metrics collection follows.

Air Force

- Software Management Indicators - 23
- CECOM STEP - 0
- Software Engineering Institute metrics - 14
- Reuse metrics - 5
- Other(11) - process, product and management metrics from many sources, using SASET tool; various management metrics; peer review metrics; performance metrics(2); SLOC; program-specific metrics; ad hoc metrics/measures; internally-required metrics; manhours, costs and lines of code; and basic productivity metrics.

One person mentioned that he is currently staffing the implementation of the Air Force metrics policy, which may ultimately result in a more systematic approach toward organization-wide metrics collection and analysis.

Army

- Software Management Indicators - 9
- CECOM STEP - 4
- Software Engineering Institute metrics - 2
- Reuse metrics - 4
- Other(6) - size, quality (using AdaMAT), COTS dependencies, code redundancy; contractor-developed metrics; tailored CECOM metrics; DA STEP metrics(2); and local SQA-defined cleanroom productivity metrics.

Navy

- Software Management Indicators - 5
- CECOM STEP - 0
- Software Engineering Institute metrics - 4
- Reuse metrics - 2
- Other(2) - function points/SLOC; LOC, manhours.

NASA

- Software Management Indicators - 1
- CECOM STEP - 0
- Software Engineering Institute metrics - 0
- Reuse metrics - 1
- Other(1) - development activities for each application.

7. Are any cost analysis, prediction or planning models used to assess and manage software technical risk?

The following cost analysis, prediction or planning models are currently in use: REVIC(13), COCOMO(15), COCOMO(COSTAR)(1), CMM(1), Timeline(3), SECOMO(3), SEER(3), GSFC Software Engineering Lab Model(1), SLIM(4), SEER-SEM(1), SPQR(2), PRICE-S(2), internally-developed models(3), APMSS(1), MicromanII(2), SASET(2), CA-Super Project(1), IDEF(1), a Logistics Cost Model (LCM) on PDSS(2), Checkpoint(1), historical model(1), WBS.SW(1), DA STEP(1), and SEER/REVIC(1).

The following results illustrate the extent of planning models utilized, as well as each model's usage frequency (in order of relative frequency) within the services.

Air Force

- Using models - 28
- Not using models - 32

Models being used:

- COCOMO - 11
- REVIC - 7
- SEER - 3
- Timeline - 3
- SASET (Software Architecture Sizing Estimating Tool) - 2
- MicroMan II - 2
- LCM (Logistics Cost Model) - 2
- SPQR - 2
- PRICE-S - 2
- Internally-developed process models - 2
- APMSS - 1
- CA-Super Project - 1
- CMM - 1
- Checkpoint - 1
- SLIM - 1
- Historical model - 1
- IDEF activity, process, dynamics and information models - 1

Army

- Using models - 9
- Not using models - 9

Models being used:

- COCOMO - 3
- SECOMO - 2
- REVIC - 1
- WBS.SW - 1
- DA STEP - 1
- SEER/REVIC - 1

Navy

- Using models - 8
- Not using models - 10

Models being used:

- SLIM - 3
- COCOMO - 1
- COCOMO (COSTAR) - 1
- REVIC - 1
- SEER-SEM - 1
- SECOMO - 1
- Internally-developed model - 1

8. What are your most pressing concerns relative to current software development practices?

Respondents' most pressing concerns regarding current practices include the following: high costs(57), low productivity(37), late delivery(64), other(41).

Air Force respondents' most pressing concerns were late delivery (34), followed closely by high costs (30), and finally by low productivity (15). However, these respondents cited 22 other problems that have also caused concern about software development and maintenance practices, and these may provide additional insight into continuing difficulties that must be successfully overcome or at least addressed in part to facilitate the path for software reuse. Some more significant concerns include: poor quality (cited by five respondents); software acquisition not yet predictable in terms of cost, schedule and quality; requirements volatility (mentioned by three persons); process management issues, including documentation, formal testing, estimation of effort and scheduling; restrictions of Government contracting law on program management action; lack of personnel resources; funding (mentioned by three people); lack of knowledge regarding good engineering practices; quality suffers for the sake of schedule; need to raise maturity level, and therefore become less reliant on most talented personnel; high maintenance costs; and lack of performance criteria.

Army respondents indicated relatively consistent concern about each of the issues listed. Late delivery was slightly higher (14) than high costs (11), followed by low productivity (10). Other concerns cited duplicated the Air Force concern about quality issues, and also highlighted the issue regarding completeness and quality of the end product vs. the user's requirements. Other concerns that emerged dealt with the following: lack of an institutional SEE; inadequate scope definition; tight schedules, and the lack of funds for development.

Navy respondents mirrored those of the Army in terms of frequency of concern expressed about the listed issues. Their greatest concern was late delivery (14), and both high costs and low productivity followed with 12 each. Additional concerns indicated by respondents include: acquisition policies/procedures; unstable budget; non-open systems; reuse; counterproductive standards; poor requirements; short development cycle; and resource (personnel) ceilings.

Both NASA respondents cited high costs as a primary concern, and one indicated that late delivery was also problematic. One person mentioned that planning and understanding requirements was an additional concern that has emerged within the NASA community.

Finally, respondents who provided no indication of organization, or who fit into no service category, responded as follows. High costs led with 7, late delivery followed with 6 and low productivity came in with 2. Other problems cited by these persons include two quality (iffy or uneven) issues, two requirements matters (good definition of requirements, and the ability to satisfy requirements), and money available for support.

Many of these additional issues of concern are no surprise to the reuse community. However, the frequency with which these matters continue to re-emerge as problems and plague the services through lack of resolution highlights the continuing requirements for vigilance. It would behoove those persons within the services with decision-making power to take action to mitigate or resolve some of these issues. Such action could defuse explosive situations before they are touched off. In some way or another, most of these unresolved issues pose great danger to the success of reuse initiatives, and therefore they should not be ignored nor their importance diminished.

8.3.2 Software Reuse

9. *Are you familiar with software reuse?*
   Yes(104),
   No(3).

Very few respondents are unfamiliar with software reuse. Only three Air Force persons expressed a lack of knowledge. We have correlated responses to this question with questions 15 and 16, which address actual reuse taking place within an organization, or a person's past experience with software reuse. We believe that such information would give the reuse community a good feel for the breadth of dissemination of the message about software reuse, as well as its overall effectiveness. Results by service of those stating familiarity with reuse,

although no reuse is taking place within their organizations: Air Force(28), Army(5) and Navy(9). Those stating familiarity with reuse, and reuse is taking place within their organizations yielded the following results: Air Force(34), Army(17), Navy(9) and NASA(2). It appears evident that the reuse community is effective in getting the message about reuse out to the larger software development and maintenance communities.

10. Do you believe that software reuse is a reasonable solution to the military services' needs for greater productivity in the face of downsizing?

Yes(80),
No(16).

Most respondents concur that software reuse is a reasonable solution; however, many caution that it is only part of the solution, not the only answer and not appropriate in all instances, and certainly not a "silver bullet". Moreover, one respondent pointed out that "reuse must be part of the overall process, not an end unto itself". Process improvement and adequate documentation are also required, and another person maintains that "the solution requires more than reuse. Actually, downsizing makes reuse more difficult, because reuse requires an up-front investment." One individual expressed the belief that reuse will require a change in management's perspective, another feels that reuse is one of many solutions that are viable, and another stated "yes - but how?" Nevertheless, only 12 Air Force, 3 Army and 1 Navy respondent disagree that reuse is a solution, either in whole or in part, while 45 Air Force, 18 Army and 17 Navy persons believe that it offers promise.

Those who disagree that reuse is a solution provided the following rationale for their opinions: haven't seen the proof yet - idea is fine, but let's watch the implementation; not while still in infant stage; no practical way of implementing on a wide scale; could be, but not as currently being practiced; and part of a much larger solution picture.

Two Army respondents provided more in-depth comments, which follow. "Downsizing (i.e., fewer resources) has always led to greater competition for resources. Competition leads to greater secrecy to maintain market advantage. Industry will guard its reusable software, and restrict its reuse under current laws. "Downsizing" for the Government means fewer individuals to "create" the needed reusable components for future systems. Reuse will require an "upsizing" of funds to accomplish. For an industry adjusting to military/defense downsizing by transitioning to commercial markets, more proprietary software will emerge. "Trade secrets" are paramount in a commercial market, and "unrestricted reuse" may vanish with the distinction between commercial and military systems as services turn to the commercial world to meet their needs. "Downsizing" of resources worldwide is leading to mergers of assets for many defense contractors. These conglomerates will "call the shots" in the downsized defense industry, including what becomes of "reuse"."

The other respondent believes that "reuse is the only way we will meet the challenges presented by downsizing. The Army needs to foster domain management, and develop, and mandate for use, reusable components which solve shared, recurring problems. My field is embedded weapons, an area which is neglected by existing reuse activity."

11. Do you believe that the highest management levels within your organization support the institution of software reuse?

Yes(62),
No(30),
Unsure(9).

For the most part, the services and NASA respondents believe that their management supports reuse. The Navy is the only exception to this general rule; their respondents indicated disagreement with this contention. A breakdown of results follows.

	Air Force	Army	Navy	NASA
Yes	39	14	7	2
No	16	6	8	0
Unsure	4	3	2	0

Air Force respondents offered several comments regarding the level of management support, even though they indicated that such support was present. Several mentioned that although management may support reuse, these persons are not well educated in reuse; others are uncertain whether their managers fully understand reuse. One person answered no to the question, because his contention was that one must understand it to support it. Finally, another person stated that although support exists, it is insufficient and not pursued suitably aggressively. A respondent at Headquarters level believes that support exists at all levels; however, he doesn't necessarily believe that this is true for domain analysis and architecture development.

12. Do you believe that barriers to the successful practice of software reuse exist within your organization?

Yes(76),
No(20).

Most respondents (45 Air Force, 18 Army and 13 Navy respondents) believe that barriers do exist. Only 14 Air Force, 4 Army and 2 Navy respondents disagree. Two respondents cited barriers that they consider to be particularly important: the lack of reuse-specific education and training, and the way in which the Government accomplishes software development. Another respondent stated that most information systems are too old to consider reuse, that person's rationale for lack of reuse.

13. To your knowledge, are there any reuse advocates (champions) or reuse experts in your organization of whom you can ask reuse-related questions?

Yes(58),
No(41).

The majority of respondents in all the services but the Navy were aware of reuse advocates and/or reuse experts within their organizations. Advocates and/or experts are either less visible or fewer persons are available to Naval personnel, as the following data indicate.

Additionally, one Air Force person points out that although knowledgeable people are available, they may not be experts.

	Air Force	Army	Navy	NASA
Yes	37	14	6	1
No	22	7	11	1

We have correlated the results of combined responses to both questions 11 and 13, to gauge the overall extent of management support. Resulting data follow, as well as some potential explanations for these results.

Extent of Management Support (Correlation)

	Air Force	Army	Navy	NASA
Question 11 - Yes & Question 13 - Yes	29	11	2	1
Question 11 - No & Question 13 - No	8	2	5	0
Question 11 - Yes & Question 13 - No	9	4	5	1
Question 11 - No & Question 13 - Yes	8	3	3	0

One could conclude that those persons who answered Yes to both questions (43 respondents) most likely work within organizations in which management supports the institution of reuse. At the very least, management is proceeding in the right direction. On the other hand, those responding No to both questions (15 persons) likely face a situation where there is little or no support for reuse. The remaining situations are more problematic. If respondents answered No to question 11 but Yes to question 13 (14 persons), management within their organizations may indeed support reuse more than is commonly believed. However, those who answered Yes to 11 and No to 13 (19 respondents) may be working with management that is giving only lip service to its stated support for reuse.

14. If your organization is not reusing software, why not?

Those organizations not currently reusing software provided the following reasons for their reuse policy: no assets(17), too costly(10), legal problems(10), contractor resistance(16), and other(37). These data, broken out by service, are as follows:

Air Force

- No assets - 10
- Too costly - 6
- Legal problems - 4
- Contractor resistance - 7
- Other - 19

Those other problems that prompted policy decisions averse to reuse include: lack of knowledge/initiation costs; not prepared to do so; ignorance; too busy, apathy; not yet practical; problems finding assets; access to library; unique customer requirements; reuse must be part of software development and engineering - education and training are needed; limited value in maintenance; only work with existing systems (This person stated that there is very little opportunity for reuse in a maintenance setting, and claimed familiarity with reuse); security problems; mixed languages and multiple contractors. Also, a larger organizational scale seems appropriate for reusable simulation modelling software. This reusable software should be developed for and reused by many organizations; each "fiefdom" has its own system/database, etc., and they don't talk; language, design barriers; different embedded systems; it's so hard to determine whether a module would work for its intended purpose that most people prefer to write a new module; unique mission's COBOL doesn't facilitate reuse; and too much undocumented code.

Several of these issues highlight needs that can be met by some existing CARDS services and products. For instance, those persons citing lack of knowledge and inadequate preparation for reuse could be well served by COAR, existing CARDS documents and other CARDS franchising activities. Additionally, the reuse community is working toward making knowledge about the availability and access of assets a realization for those actively pursuing reuse; however, the CARDS team could continue to support these efforts, and ensure elimination of library access issues as barriers to those in the reuse field. The education and training issues mentioned could also provide focus for additional training courses to be developed by the CARDS team, and perhaps foster a new level of cooperation with academia to ensure that viable approaches toward reuse are integrated into both formal and external educational structures.

Finally, the CARDS team as well as the larger reuse community should be concerned about persons expressing the belief that reuse has limited value in maintenance. Although development is often the focus for our reuse efforts, and the point where many long-term and considerable benefits will emerge, reuse in maintenance also offers significant cost savings and should receive due attention. Perhaps the reuse community needs to slightly redirect the

focus of policy addressing this issue to ensure that the message about reuse benefits throughout the full extent of the life cycle is sent and received.

Army

- No assets - 2
- Too costly - 0
- Legal problems - 2
- Contractor resistance - 3
- Other - 6

Army respondents stated the following rationale for their organizations not practicing reuse: waiting for direction from headquarters; Project Manager's loss of control; management resistance; not clear how to effect in contracts; not a flashy thing to do - can't make colorful viewgraph out of it; and lack of education on reuse. One person stated that although active reuse is taking place within the organization, problems with all of the above still occur.

Navy

- No assets - 4
- Too costly - 3
- Legal problems - 3
- Contractor resistance - 3
- Other - 8

Navy respondents provided the following reasons for lack of reuse: management policy; management resistance; no incentive to do so; no one looks for it; just getting started; need cultural change; need reliable reusable components; and reorganization is needed, because there exists a lack of coordinated effort to institute software engineering practices.

Those respondents who neglected to state an organization or who didn't fit into the services categories yielded the following data:

- No assets - 1
- Too costly - 1
- Legal problems - 1
- Contractor resistance - 3
- Other - 4

Their rationale for reuse not taking place included: management hasn't a clue about it and doesn't think we'll save; no one is responsible person; application independence; and Government offices not interested.

15. Have you worked on a program in which reuse was practiced?

Yes(68),
No(45).

Apart from the Army, about equivalent numbers of personnel have and have not worked on reuse programs.

	Air Force	Army	Navy	NASA
Yes	34	17	15	2
No	27	5	13	0

16. *What type(s) of component was reused?*

The types of components being reused include the following: domain model(10), requirements(21), architecture(17), designs(28), specifications(20), code(57), test suites(19), documentation(26), other(3) - specific special routines, job control and physical resource, and engineering reference models.

These data are broken out by service in a side-by-side presentation to enable quick comparisons between services. Additionally, information about code only reuse is provided to help determine how well the message is being disseminated about the significant savings possible through extensive reuse of a wide range of life cycle components, and whether widespread reuse is being implemented within organizations. According to the data, there is little code only reuse taking place; code reuse is usually paired with reuse of other assets that may significantly augment potential savings.

Breadth of Software Reuse

Reused Components	Air Force	Army	Navy	NASA
Domain Model	4	4	0	2
Requirements	8	9	2	2
Architecture	8	6	1	2
Designs	12	9	5	2
Specifications	8	8	2	2
Code	30	15	10	2
Test Suites	10	6	2	1
Documentation	15	7	3	1
Other	3	0	0	0
Code Only	7	0	2	0
No Reuse	30	5	8	0
Total Respondents	65	22	18	2
Percentage of organizations practicing reuse	54	77	56	100

17. Were Commercial-Off-The-Shelf (COTS) or Government-Off-The-Shelf (GOTS) components used?

Yes(46),
No(17).

A breakdown of these results by service/NASA follows.

	Air Force	Army	Navy	NASA
Yes	22	13	10	1
No	11	3	2	1

Within the Air Force, there were 11 instances of reuse that involved neither COTS nor GOTS use. All but three cases of Army reuse and two cases of Navy reuse used COTS or GOTS as part of their reuse strategy.

18. Was there a set of minimum criteria used to evaluate potential reusable components?

Yes(26),
No(31).

Both the Army and Navy respondents indicated greater usage of such criteria than lack of use, but the Air Force results indicated otherwise. Only 10 Air Force respondents stated that such criteria are used, whereas 21 do not use any criteria. 10 Army respondents indicated that a set of criteria is used, while only 6 do not make use of such criteria. Navy results were similar; 6 respondents use minimum criteria, but 4 do not.

19. Was reuse pursued with components originally designed for future reuse, were components reengineered to make them suitable for reuse, or other (please explain)?

Although a considerable amount of reuse has taken place with components originally designed for future reuse, including COTS and GOTS (Air Force - 16, Army - 6 and Navy - 7), much reengineering has also occurred (Air Force - 19, Army - 11 and Navy - 4). In addition, Air Force respondents cited the following additional instances of reuse: application interfaces built to support COTS interchange; reengineering in which software from a newer version aircraft was placed into an older version for the sake of compatibility; reverse engineering; and one instance of reengineering in which a respondent who knew the code well hacked it to fit the new application.

20. In your opinion, were the components reliable and of high quality?

Yes(48),
No(19).

The majority of respondents indicated that, for the most part, components were reliable and of high quality. However, several Air Force and one Army person answered both Yes and No to the question, since component quality was inconsistent. One Army respondent stated that his

answer was Yes on COTS, but No on GOTS. The services/NASA responded as follows:

	Air Force	Army	Navy	NASA
Yes	24	14	8	2
No	13	4	2	0

21. Was any prototyping of the system conducted at any point during the life cycle, either to help refine requirements or test the applicability of potential reusable components? No

Prototyping is taking place within each of the services and NASA, with most targeted toward the refinement of requirements. Breakdowns of applicable data follow.

	Air Force	Army	Navy	NASA
Refine requirements	15	9	7	1
Test applicability	11	4	4	1
No	13	3	1	0

22. Did you encounter significant problems in: using as is, modifying, reengineering or integrating these components?

Using as is(25),
Modifying(16),
Reengineering(12),
Integrating(19).

The services and NASA encountered significant problems in the following areas in utilizing assets:

	Air Force	Army	Navy	NASA
Using as is	15	5	3	2
Modifying	11	3	1	1
Reengineering	9	1	2	0
Integrating	9	5	4	1
Reuse taking place, but no significant problems cited	11	8	5	0

23a. Are any cost analysis or prediction models used to analyze the impact of software reuse on projected programs?

Yes(10),
No(46).

23b. If yes, please specify the model(s) used.

Few such models are used to determine what impact software reuse has had on programs. Between the three services, only 10 respondents indicated such use, with the Army leading the list with 5 respondents. Only a few more Army respondents stated a lack of model usage (8). Those models being used include: REVIC(3), COCOMO, SLIM, SoftCost, If-Then-Else, and the Reengineering Economics Handbook. 46 respondents stated that no models are being used to gauge the impact of reuse. Please note the dichotomy between the results of this question and that of question number 7. Although there is widespread use of models to manage software technical risk, these models [illegible] to be consistently applied to the subsequent analysis of the impact of software reuse.

24. Did reuse increase productivity?

Yes(40),
No(6).

For the most part, respondents believe that reuse did increase productivity. However, many expressed reservations, or responded yes with a caveat or explanatory message. One Navy respondent stated decisively that productivity was definitely increased. Other respondents were less emphatic in their statements. Several questioned whether their organizations were able to ascertain whether productivity was affected, pointing out that they don't know how one properly measures productivity. One person also cited the lack of reuse metrics to enable someone to make such a determination. However, they did concede that reuse resulted in reduced errors and time required to accomplish tasks. Another respondent stated that productivity improvements were not initially observed, but occurred only after a repository of reusable software was established. Finally, two others stated that productivity improved only in certain circumstances, and another believes that reduction of work through reuse ultimately resulted in improvement.

25. How would you describe your software reuse experience(s):

Extremely positive(11),
Somewhat positive(33),
Neither negative nor positive(17),
Somewhat negative(0),
Extremely negative(1).

The services yielded relatively similar results in response to this question:

	Air Force	Army	Navy	NASA
Extremely positive	6	2	2	1
Somewhat positive	14	10	8	1
Neither positive nor negative	12	4	1	0
Somewhat negative	0	0	0	0
Extremely negative	1	0	0	0

In general, many questionnaire recipients respond if their experience has been particularly positive or especially negative. Although we cannot assume that few persons have actually encountered negative experiences with reuse, it is heartening to see evidence of some positive experiences, or at the very least the lack of a general negative impression.

8.3.3 Recommendations

All of the recommendations will be addressed by service.

26. Is there anything you would do differently next time in either reusing available components or developing reusable components?

Air Force respondents provided the following comments: currently making reuse components available to contractor - would like to make their use mandatory in future; more up-front analysis and design for reuse; definition of reuse is vague; develop reusable packages earlier in program; develop for reuse, rather than trying to fit the square peg in a round hole; learn their original requirements; make as generic as possible for portability; greater emphasis on designing for reuse; wider use; up-front domain analysis; assess the extent of reuse planned. Clarify contractor's description of reuse; design reuse from the start using object-oriented language; follow a pre-defined process and a stable architecture; enforce wider reuse; we're providing a plan for transitioning to Ada 9X; be selective and more specific to detail requirements; and prepare/train developers to engineer reuse components.

Army comments included the following: better documentation/test; take away all funding from users and put in central line control; reused components should have been originally designed for future reuse purposes; demand more insight into contractor IR&D; need more testing and drivers to adequately test components from other sources; and increase use of Cleanroom Software Engineering.

The following comments were made by Navy respondents: expand scope to include unit test in library; begin reuse earlier in the life cycle; assure that components to be reused are of high quality; obtain support from original developers; and spend more time researching sources.

Comments from our miscellaneous sources recommended that components be developed to be system independent, and that a domain analysis be accomplished.

Those who fit into the miscellaneous category provided these comments: domain analysis, and write components so that they are system independent.

27. Based upon your software reuse experience, what do you feel must happen or work out right for reuse to succeed? Where would you hate to see something go wrong?

Air Force respondents provided the following comments in regard to this query: must have full confidence in reuse components and understand legal and contractual requirements; must pre-plan and design for reuse; very modular construction with clearly defined input/output and

clearly documented function; careful attention to details; specifications and design must be right; more standard software engineering practices; design components for reuse, and provide database list of reusable components; products must be designed/coded for reuse; better front-end planning; define a good stable architecture; need a widely available source and people trained to use it; need common units and variable types for input/output between components; need bigger libraries, need design/coding for reuse; object-oriented languages with inheritance, like C++; start early; larger-sized assets and earlier in life cycle; Government cannot force reuse; management commitment; new code must be set up with reuse in mind; reuse repository with certification level facilities via Internet; software, designs, etc. must be done with potential reuse in mind; metrics are necessary and must be used consistently; specific mapping of capability to requirements; 50% increase and reuse metrics implementation; and doing reuse without consciously knowing that they are in a "reuse program". It should develop at the grass roots level, and come from the bottom up, with top-level support, of course.

Where Air Force respondents would hate to see something go wrong: no funding nor interest; need to work reuse plan first before this can be answered; during system test and fielding; anywhere; and regarding legal implications for cross-company use.

Army personnel responded as follows: Government (Army) take lead and responsibility for units. Mandate known good products.; Requirements must be well-identified. Must understand functionality of reuse software.; moderation; make reuse libraries very accessible and easy to use; it must be part of PM's mission and requirements; cooperation; must be fully tested and documented; must use Ada; need to make investment in domain analysis and architectural analysis; legal issues; contractor resistance; to be planned for and enforced by the Project Manager; need to identify reuse packages earlier in life cycle; legal issues need to be worked out and incentives for contractors; be realistic; "Reuse champions need to make sure that personnel within DoD do not equate reuse with code reuse. Everyone seems to think of reuse libraries when they hear reuse. Domain model, requirements, and architecture reuse are much more important concepts."

An Army respondent provided a more in-depth expression of his views regarding some of these issues. "Legal issues are too often overlooked as technical issues become an attractive intellectual exercise. Economic barriers are also understated, and impacts discounted. It's tempting to suggest that the Government must "upsize" its technical expertise by hiring the laid-off contractor software expertise. If DoD wants reuse to succeed, it must bring the expertise in house, I believe. The biggest mistake is for DoD to think that it is exempt from these factors necessary for success of reuse, with too high expectations."

Army respondents would like to avoid insufficiently robust code and design, and unreasonable expectations.

Navy respondents cited the following recommendations: legal and financial incentives to break domain boundary; upper management support/funding; there has to be a dedicated plan and management metrics must be captured throughout the duration of the program; maturity of reuse libraries - consistent look and feel; and paying for components.

Those respondents who fit into the miscellaneous category offered the following insight into what must work out right: pre-planned reuse, and configuration management. One respondent would hate to see something go wrong at test time.

28. What services could the software reuse community provide to best assist you in meeting the objectives of software reuse?

Air Force respondents expressed the following opinions regarding reuse services: component availability is great, but how do I implement reuse contractually and programmatically; integrated access to all repositories, and GUI front-end; make information easy to get to, and make the search worth it; better library search; easy search capabilities; let me know what software is available for reuse; information; simple listings of components would be a start; storage of a reuse library; establish criteria definitions and public reuse repositories; reuse repository with certification level facilities via Internet; establish a DoD point of contact that I could call for basic information; reuse service similar to CompuServe, in which "tested", "high quality" modules are easily available, with all supporting documentation included; better guides, technically and legally; reduce the Not Invented Here (NIH) syndrome; good ways to design and evaluate reusable components; standardized criteria, database implementation; models for domain architectures in the embedded avionics domain; knowledge, training, tools; incentive model and success stories; awareness and advocacy; specific training on reuse engineering design strategies - "design for reuse"; and management and engineering training services.

One respondent hopes that her research will help the software reuse community.

Army respondents provided the following insight into necessary services: financial incentives to contractors; develop specific components to solve common problems; an easy method to identify related software products of quality; availability; familiarization, training and marketing; be helpful! advertise in well-known Federal or commercial newspaper; address Ada 9X issues; guidelines; training; and software testing is much more costly than code development, but does not get required attention.

Navy respondents answered as follows: freeware from D.B. in standard template; market success; follow Free Software Foundation; and access to a reusable component directory by platform and library.

Those who fit into the miscellaneous category offered the following opinions: publish standard domain-specific architectures; make reuse libraries easily accessible and user-friendly; and provide domain analysis methods and services.

29. Do you have any recommendations that could improve the software development or maintenance process in your organization?

Air Force respondents provided the following recommendations: more automated tools for the whole process; paradigm shift in design, development methodology, and better defined development processes; constant effort with ongoing initiatives, managers who are committed

to change and improvement; heading for higher SEI levels; training engineers in software engineering; software professionals must better understand software engineering; establish software engineering processes and put them in friendly, easy-to-use, software development workstations; standardization; more training in software engineering; need better requirements analysis and system-unique models for cost/effort estimating; perform Enterprise (domain) analysis across ALL supported systems; let the contractor do the work - keep Government out of their way; reengineer badly developed code (all of it - not just operational), and institute formal reuse into our process; follow the Systems Engineering Process as taught by DoD; more people; document processes; get a handle on the legal stuff; take the time to look at reuse first!; overall process improvements along SEI CMM guidelines; software development can be improved by "growing" software engineers; process standardization, and management buy in; more usage of COTS, particularly LAN-based DBMS; fewer interruptions in daily schedule - meetings, details, etc.; training on CASE tools; and providing adequate funds to support the right people.

One Air Force respondent provided an in-depth process-related comment. "I believe the true improvements will come from a Software Process Improvement Program (SPIP) as opposed to "reuse" or "metrics" programs as stand-alone improvements. In order to successfully implement reuse, I believe an organization must achieve CMM Level 2 in Software Configuration Management as a minimum. Mandating reuse or metrics without institutionalizing the Management KPAs in the CMM will probably result in a large amount of wasted time. As a case in point, AFGWC has a SPIP in place, with strong senior leadership support. There are many other Air Force organizations that could learn from the AFGWC example." Two other respondents stated that their Software Engineering Process Groups are constantly working on any and all recommendations that are received, and an additional person stated that his group is currently working on the development of processes to improve software development. Finally, one respondent with a less positive experience expressed her frustration that although a report was prepared and presented to management after a SEI exam, management didn't like what was presented.

In another vein, one person stated that "a good place to start would be a DoD-wide standard language for simulation modelling (e.g., Monte Carlo models, especially discrete-event type). I tend to think that Ada would be a good choice; in other words, the simulation standard would conform with the DoD standard for software in general."

Another Air Force person stated that it is "better to do things in a more gentle way". His organization is trying to reach SEI level 2, and his main recommendation is to "develop spontaneously at pockets of grass roots level". Although this may be a difficult task, requiring much effort, one major benefit is the decrease in resistance to reuse due to its lack of formalism.

Army respondents offered the following recommendations: realistic, practical training for managers and staff; establishment of institutionally standardized SEE and PDSS policies; education, demonstration; provide education on the software engineering principles needed by both contractors and organizations monitoring contractors; resources, time and people; cooperation - availability of original software authors to ask questions or provide assistance in

integrating process; use Ada 9X; initiate a software reuse program; delivery date should be set to "time to develop", not staff level desires; and need to overhaul the way the DoD/Government does software development and procurement.

One Army respondent stated that his organization is currently working to improve their process, as is one Navy respondent.

Navy respondents provided the following recommendations: define system architecture, install software metrics methods; better configuration management; establish a point of contact and small staff to organize, research, develop, maintain, distribute and enforce use of reusable assets; more complete systems engineering (requirements gathering) up front!; and need higher-level incentives.

Respondents who fit into the miscellaneous category provided the following recommendations: access to list of reusable software; management needs to stand behind it and allow the reuse of software; use software engineering technology (open SEEs) for maintenance, to include process enactment; upper management needs to be convinced to incorporate reuse, and money needs to be available for support of software development and maintenance.

Several respondents provided opinions and concerns about reuse and current policy that not only affect their organizations, but impact the services or the DoD as well. These comments follow. "Reuse should start with requirements laid out with the thought of using off-the-shelf components, instead of providing requirements that force manufacturers to write their own GUIs, for instance. For real reuse, the Government should mandate C++ instead of Ada, since the commercial world is generating a lot of off-the-shelf, tested, documented, maintained and cheap software components." Another respondent stated that "Too many programmers and developers use the academic approach: "prove it can run". What is needed is "prove it always runs". Testing and software maturity, exception testing, etc. do not get funding and schedule to do the job right. All too often, early milestones are met at the expense of later ones (e.g., testing)." One person provided details about an attempted switch to Ada and software engineering principles about 18 months ago. They encountered significant obstacles, including the unwillingness of high-level managers to extend milestones to facilitate good designs, and the resistance of programmers to change from their "C, C++" ways. Finally, another respondent stated that "some at DoD are pushing reuse as a concept without giving clear guidance to the field as to what they are being asked to do. Any action which would improve this situation would be helpful."

31a. Do you think that the DoD needs a software advocate and/or reuse advocate?

Yes(97),
No(12).

31b. If yes, software advocate, reuse advocate, or both?

Software advocate(11),
Reuse advocate(8),
Both(77).

Study respondents overwhelmingly support the establishment of a software and/or reuse advocate. Nevertheless, concerns were expressed by several respondents. An Air Force respondent believes that reuse must be worked under the umbrella of Software Process Improvement Programs. "Since about 75% of all software organizations are at CMM Level 1, it's very doubtful that they'll be able to implement reuse. Get the organization to improve its capability before trying to force reuse down its throat!" Other respondents also expressed concerns about the already massive bureaucracy in place, and wondered whether an organization that could handle these tasks already exists.

Respondents differ greatly in terms of which organizational levels they believe should be represented by advocates. Several respondents emphasize the necessity of encouraging advocacy at all levels, since they believe this will be necessary to ensure success. Those at the highest levels could impress on lower management levels that necessary procedures will be enforced, but the person selected would have to be someone with the authority to provide enforcement capability and funding within the DoD. One respondent stated that the advocate should only emphasize the importance and lead, not dictate policy and standards. Another emphasized the importance of establishing a grass roots committee, since anyone else rapidly loses touch with what goes on in the organization. Finally, another respondent believes that the individual selected should be a technician, not a politician!

General consensus appears to be that advocacy is required at all levels - at the highest levels to ensure provision of enforcement, funding and necessary leadership, but also at each level within all organizations down to the lowest grass roots levels, to ensure information flows in both directions, thereby maintaining each high-level advocate's touch with the pulse of the organization.

9 FUTURE EFFORTS

The data and results reflected in this document are available in dBase, and accessible to members of the CARDS team for future technology transition and franchising activities. They may be used to identify candidate organizations in need of assistance, and should appropriately target CARDS efforts toward those activities that offer the greatest potential to address actual and perceived needs within the DoD.

REFERENCES

[1] Market Study, Central Archive for Reusable Defense Software (CARDS), STARS-AC-04118/001/, 30 June 1993.

[2] Market Study, Central Archive for Reusable Defense Software (CARDS), STARS-AC-B001/000/00, 30 September 1993.

[3] Market Study, Central Archive for Reusable Defense Software (CARDS), STARS-VC-B003/00, 23 December 1993.